Closed-Loop Upgrade of a 5-DOF Robotic Manipulator

Shannon Edwards

Supervisor: Dr. Patrick Keleher

A thesis submitted in partial fulfilment of the requirements for the degree of Bachelor of Engineering (Electrical)

Faculty of Sciences, Engineering and Health

CQUniversity

June 2012
Abstract

Typically robotic manipulators, such as the well-known robotic arm, utilise electric motors to actuate the joints. Brushless DC motors are becoming increasingly popular in robotic applications due to numerous advantages and recent advancement in power electronics technology, allowing for high power devices with precise positioning capabilities. This thesis presents an investigation into the refurbishment and upgrade of a Mitsubishi Movemaster RM-101 robot that was recovered from storage at the CQUniversity Rockhampton campus. The primary goal of the project was to upgrade the unit to allow for closed-loop position control. Brushless DC motors with quadrature-type position encoders were installed on the robot. Extensive investigation was carried out regarding brushless DC motor controllers, resulting in schematic diagrams and firmware code for a prototype motor control circuit. This thesis discusses the detailed electrical design, and also presents the possibility for new advanced functionality that can be added to the unit with the upgraded hardware.
Acknowledgements

I would like to thank and acknowledge the following people for their assistance, guidance and support throughout the duration of this project.

First of all I would like to thank my supervisor Dr. Patrick Keleher for his support and advice. I would also like to sincerely thank Mr. Ian Tomlinson for his valued input and suggestions throughout all stages of the project.

I am also grateful to CQUniversity and the Faculty of Sciences, Engineering and Health for financing the project and purchasing a number of required parts and equipment.

Finally, I wish to thank my partner Kate, friends and family for their support and encouragement.
Declaration of Originality

I certify that the ideas, designs and experimental work, results, analyses and conclusions set out in this dissertation are entirely my own effort, except where otherwise indicated and acknowledged.

I further certify that the work is original and has not been previously submitted for assessment in any other course or institution.

Shannon Edwards
Student Number: s0146725

Signature

1/6/2012
Date
## Contents

List of Figures vi

List of Tables vii

List of Acronyms viii

1 Introduction 1
   1.1 Robotic Manipulators ..................................... 1
   1.2 The Mitsubishi Movemaster RM-101 .......................... 2
       1.2.1 Operating Limits ................................... 3
   1.3 Project Objectives ........................................ 4
   1.4 Outline of Thesis ......................................... 6

2 Literature Review 7
   2.1 Direct Current Motors ...................................... 7
       2.1.1 Brushed DC Motors .................................... 7
       2.1.2 Stepper Motors ....................................... 10
       2.1.3 Brushless DC Motors ................................ 12
   2.2 Motor Drive Topologies ................................... 16
       2.2.1 Half Bridge .......................................... 16
       2.2.2 H-Bridge .............................................. 17
       2.2.3 Wye-Connection ..................................... 19
       2.2.4 Delta-Connection ................................... 20
   2.3 Position Feedback .......................................... 21
       2.3.1 Hall Effect Sensors .................................. 21
       2.3.2 Incremental Encoders ................................ 24
       2.3.3 Absolute Encoders .................................. 26
       2.3.4 Back EMF Detection ................................ 27
   2.4 BLDC Controller Design .................................... 28
       2.4.1 Inverter Switching Schemes ......................... 29
       2.4.2 MOSFET Characteristics ................................ 32
       2.4.3 Overload Protection .................................. 36
   2.5 Feedback Control ........................................... 39
       2.5.1 PID Control Theory .................................. 40
2.5.2 In Relation to Servo Motors 42

3 Hardware 44
   3.1 Design Process 44
   3.2 Selection of Major Components 45
   3.3 Circuit Implementation 48
   3.4 Controller Prototype 50
   3.5 Mechanical Modifications 52

4 Embedded Program 54
   4.1 Control Routine 54
   4.2 Interrupt Service Routine 56
   4.3 Proof of Operation 59

5 Simulink Model 60
   5.1 BLDC Motor Model 60
   5.2 Three-Phase Inverter 63
   5.3 Results 66

6 Discussion 68
   6.1 Summary 68
   6.2 Further Work 69

References 72

Bibliography 75

A Schematic Diagrams 76

B Firmware 79

C Nanotec DB42S03 datasheet 87

D Fourier expansion of trapezoidal back EMF 89
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Movemaster RM-101 nomenclature (Mitsubishi Electric Corporation 1982)</td>
<td>2</td>
</tr>
<tr>
<td>1.2</td>
<td>Home position of the RM-101 (Mitsubishi Electric Corporation 1982)</td>
<td>5</td>
</tr>
<tr>
<td>2.1</td>
<td>Cutaway of a permanent magnet brushed DC motor (Sclater &amp; Chironis 2007)</td>
<td>8</td>
</tr>
<tr>
<td>2.2</td>
<td>Exploded view of a hybrid stepper motor</td>
<td>10</td>
</tr>
<tr>
<td>2.3</td>
<td>Typical open loop response of a stepper motor (Athani 2007)</td>
<td>11</td>
</tr>
<tr>
<td>2.4</td>
<td>Stepper motor mid frequency oscillation (Athani 2007)</td>
<td>12</td>
</tr>
<tr>
<td>2.5</td>
<td>Cutaway view of a brushless DC motor</td>
<td>13</td>
</tr>
<tr>
<td>2.6</td>
<td>Ideal phase back EMF and current waveforms for a BLDC motor</td>
<td>14</td>
</tr>
<tr>
<td>2.7</td>
<td>Ideal phase back EMF and current waveforms for a BLDC motor</td>
<td>15</td>
</tr>
<tr>
<td>2.8</td>
<td>Half bridge drive topology (Hanselman 2006)</td>
<td>17</td>
</tr>
<tr>
<td>2.9</td>
<td>H-bridge drive topology (Hanselman 2006)</td>
<td>18</td>
</tr>
<tr>
<td>2.10</td>
<td>Y-connection topology (Hanselman 2006)</td>
<td>19</td>
</tr>
<tr>
<td>2.11</td>
<td>Delta-connection topology (Hanselman 2006)</td>
<td>20</td>
</tr>
<tr>
<td>2.12</td>
<td>Principle of the Hall Effect sensor (Wilson 2004)</td>
<td>22</td>
</tr>
<tr>
<td>2.13</td>
<td>Relationship between BLDC phase currents and Hall sensor outputs</td>
<td>23</td>
</tr>
<tr>
<td>2.14</td>
<td>Quadrature outputs showing phase shift and binary transitions</td>
<td>24</td>
</tr>
<tr>
<td>2.15</td>
<td>Gray coded (a) and binary coded (b) absolute encoder wheels</td>
<td>26</td>
</tr>
<tr>
<td></td>
<td>(Borenstein, Everett &amp; Feng 1996)</td>
<td></td>
</tr>
<tr>
<td>2.16</td>
<td>Typical brushless DC motor controller example</td>
<td>29</td>
</tr>
<tr>
<td>2.17</td>
<td>Inverter hard chopping</td>
<td>30</td>
</tr>
<tr>
<td>2.18</td>
<td>Inverter soft chopping</td>
<td>31</td>
</tr>
<tr>
<td>2.19</td>
<td>Inverter soft chopping 60° variation</td>
<td>32</td>
</tr>
<tr>
<td>2.20</td>
<td>Cross sectional view of N-Channel MOSFET showing various inter-junction capacitances (Pathak 2001)</td>
<td>33</td>
</tr>
<tr>
<td>2.21</td>
<td>Basic bootstrap power supply circuit</td>
<td>35</td>
</tr>
</tbody>
</table>
**LIST OF FIGURES**

2.22 High and low-side resistive shunt measurement (Lepkowski 2003) ........................................... 37
2.23 Combined offset and noise reduction circuit for low-side resistive shunt measurement (Lepkowski 2003) ........ 38
2.24 Block diagram of a basic feedback loop (Åström 2002) .... 40
2.25 Block diagram of servo motor with cascaded position, speed and current controllers .............................. 43

3.1 Nanotec DB42S03 brushless DC motor (a) and with encoder mounted (b) ........................................ 46
3.2 Block diagram of BLDC controller ........................................ 49
3.3 Block diagram of circuit power stage ................................. 49
3.4 Prototype of motor control circuit on breadboard ............... 51
3.5 Reconstructed Movemaster RM-101 ................................. 53

4.1 Main routine ......................................................... 55
4.2 Proposed control strategy ........................................... 57
4.3 Interrupt service routine ........................................... 58

5.1 Simulink model of BLDC motor and 3-phase inverter in open-loop configuration ...................................... 60
5.2 Equivalent circuit for delta-connected brushless DC motor .......................... 61
5.3 Contents of BLDC motor block .................................... 63
5.4 Calculation of position and sine wave vector in Subsystem .. 64
5.5 Equivalent circuit for delta-connected BLDC motor during commutation state AB .......................... 64
5.6 Velocity response of delta-connected BLDC motor model .......................... 67
5.7 Torque response of delta-connected BLDC motor model .......................... 67

A.1 Power stage schematic diagram .................................... 77
A.2 BLDC controller schematic diagram ............................... 78
# List of Tables

1.1 List of motors and associated joints and limbs .................................. 3
1.2 Rotational operating limits of the RM-101 ........................................ 4

2.1 Brushless DC commutation logic ....................................................... 22
2.2 Quadrature decoding sequence ......................................................... 25
2.3 Comparison of current sensing methods (Lepkowski 2003) .................. 39
2.4 Controller gains as per the Zeigler-Nichols tuning method ................. 42

3.1 Calculation of required power supply unit ....................................... 48

5.1 Inverter output voltages ................................................................. 66
5.2 Simulink simulation parameters ...................................................... 66
List of Acronyms

ADC  Analogue-to-digital converter
BLDC  Brushless DC
DOF  Degree of freedom
EMF  Electromotive force
EMI  Electromagnetic interference
HMI  Human machine interface
IC  Integrated circuit
I²C  Inter-integrated circuit
ICSP  In-circuit Serial Programming
IGBT  Insulated Gate Bipolar Transistor
I/O  Input/output
ISR  Interrupt service routine
MMF  Magnetomotive force
MOSFET  Metal-Oxide Semiconductor Field-Effect Transistor
NdFeB  Neodymium-iron-boron
PM  Permanent magnet
PWM  Pulse width modulation
QEI  Quadrature Encoder Interface
Chapter 1

Introduction

1.1 Robotic Manipulators

Generally speaking, a robot arm consists of mechanical links that rotate about a joint. Each joint that can be independently actuated is called an axis or degree of freedom (DOF). A robot arm with two freely movable joints would therefore be referred to as a 2-axis or 2-DOF unit. Small robotic arms, such as the one used for this project, are usually actuated by DC motors, with the most common being stepper and brush-type DC motors. Brushless DC motors are also becoming more popular as the cost of power electronics and strong rare-earth magnets decrease.

Robotic manipulators have many uses. Perhaps the most obvious is the pick-and-place operation, where the unit is designed to grab, lift, transport and/or position objects. The Mitsubishi Movemaster RM-101 used for this project is a pick-and-place type robot. Positioning applications such as pick-and-place operations have a few general requirements, which are outlined below:

- **Torque.** Continual rotation application, such as fans and pumps, may prefer high speed rather than high torque, depending on the load requirements and whether or not the device is allowed to start up slowly. Positioning devices however undergo continual acceleration and deceleration as the arm moves from one position to another, and therefore high torque is preferable.

- **Feedback.** As a minimum, position feedback is required so that the device knows where its end effector is located. A positioning device would not be very effective if it did not know where it was positioning its payload. As will be discussed later in Section 2.5.2, having a position transducer also allows for speed control, which can be used to implement complex velocity profiles as the device moves from one position to another.
• **Control.** The device should have the ability for a human to input desired outcomes such as speed or position. A control element is needed to take the sensor input and automatically adjust the motors’ operating parameters such that the device performs as expected or programmed.

### 1.2 The Mitsubishi Movemaster RM-101

The Mitsubishi Movemaster RM-101 is a basic 5-DOF robotic manipulator that was produced in the early 1980s as a low-cost solution for hobby and educational robotic applications. As is common with similarly styled robots, each link of the manipulator is identified by the equivalent joint or limb of a human arm. The joints are referred to as the shoulder, elbow, wrist and waist, which connect the upper arm, forearm, hand, body and finger limbs.

![Figure 1.1: Movemaster RM-101 nomenclature (Mitsubishi Electric Corporation 1982)](image)

To actuate the joints, the original design incorporates six DC stepper motors which are identified by the prefix ‘M’ and an incremental number. Table 1.1 provides an overview of the six motors and the respective joints and limbs that they actuate.

Most of the joints are actuated mechanically via gear, chain-drive or pulley configurations. The waist and shoulder joints are both operated via a gear configuration. The output shafts of motors M1 and M2 are connected to large cogs, such that when the motors are energised and rotate, the respective joints are also rotated. The elbow and wrist joints also employ...
gears to rotate the respective limbs, however these joints are coupled to the motor shafts via chain-driven sprockets.

The wrist is unique in that it uses two motors and a bevel gear joint, and it is because of this configuration that the joint can be made to rotate about two axes. Rotating both wrist motors M5 and M6 in the same direction results in the hand rotating clockwise or counter-clockwise about the x-axis (looking face on from the front of the robot). If the motors are rotated in opposing directions, the wrist motor will pivot up or down; and finally, if the motors are rotated in opposing directions and at different speeds, the wrist will both pivot and twist.

Finally, the fingers are opened or closed by a pulley configuration. As M6 is rotated, the pulley wire is placed under tension and the finger grips close. Rotating M6 in the opposite direction releases tension on the pulley wire, subsequently opening the hand.

### 1.2.1 Operating Limits

Due to the mechanical design of the RM-101 manipulator and obstruction from other limbs, not all joints on the unit can be rotated through a full 360° revolution. The instruction manual specifies a number of operating limits. These limits are listed in Table 1.2.

There are a few things to note with the values listed in the table below; the wrist motors in particular deserve special consideration when determining the operating limits. As the wrist is manoeuvred by operating two motors rather than one, the angular displacement of both motors must be considered.

The instruction manual for the RM-101 specifies two equations that determine the ranges of the wrist motors M4 and M5. As the original design employs stepper motors, these two equations are, however, expressed as functions of the number of steps the motors are moved through, rather than the angular displacement of the motor shafts. Knowing the step angle of the original motors though, these equations can easily be translated to a function of angular displacement. The limits of the two wrist motors are now determined by the following two expressions:

<table>
<thead>
<tr>
<th>Motor</th>
<th>Joint</th>
<th>Limb</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1</td>
<td>Waist</td>
<td>Body</td>
</tr>
<tr>
<td>M2</td>
<td>Shoulder</td>
<td>Upper arm</td>
</tr>
<tr>
<td>M3</td>
<td>Elbow</td>
<td>Forearm</td>
</tr>
<tr>
<td>M4</td>
<td>Wrist (left)</td>
<td>Hand</td>
</tr>
<tr>
<td>M5</td>
<td>Wrist (right)</td>
<td>Hand</td>
</tr>
<tr>
<td>M6</td>
<td>—</td>
<td>Finger</td>
</tr>
</tbody>
</table>

Table 1.1: List of motors and associated joints and limbs
Motor | Joint | Operational angular coverage |
---|---|---|
M1 | Waist | $240^\circ$ ($\pm 120^\circ$) |
M2 | Shoulder | $150^\circ$ ($\pm 75^\circ$) |
M3 | Elbow | $120^\circ$ ($120^\circ$ forward, $45^\circ$ back) |
M4, M5 | Wrist (bend) | $180^\circ$ ($\pm 90^\circ$) |
M4, M5 | Wrist (rotate) | $360^\circ$ |

Table 1.2: Rotational operating limits of the RM-101

\[
Rotation : -360^\circ \leq 0.5(m_4 + m_5) \leq 360^\circ \quad (1.1)
\]

\[
Bend : 0.5|m_4 - m_5| \leq 90^\circ \quad (1.2)
\]

Where $m_4$ and $m_5$ are the angular displacements of the wrist motors. The expressions indicate that for rotation or bending, the wrist output position will be determined by one half of the combined angular displacement of the two motors. (The reader is directed to page 9 of the RM-101 instruction manual for the original wrist expressions.)

Another point to note when referring to Table 1.2 is that the limits in parenthesis are the maximum degrees of displacement allowable from the home position. The home position is a predefined position prescribed by six points on the robot marked by a HP symbol. Actuating the joints so that all these points are aligned puts the unit into its home position (see Figure 1.2).

### 1.3 Project Objectives

The objective of this project was to refurbish and upgrade a Mitsubishi Movemaster RM-101 robotic manipulator that was recovered from storage at the CQUniversity Rockhampton campus. The original design of the Movemaster RM-101 is an open-loop system implemented with DC stepper motors, which were controlled by commands sent to the robot over the parallel port of a personal computer. The user would input a command into a terminal application running on the computer. The command would then be transferred to the robot, which would decode the message and perform the desired operation. The commands used for controlling the robot were alphanumeric ASCII strings, usually prefixed by a letter indicating the operation to be performed and then a numerical value.

Without feedback, the original communication process was one-way. The user would send a command to the robot and assume that the desired output had been achieved. The addition of position feedback would allow for verification of the output. For use as a teaching aid, it would also be desirable to receive information such as angular position and velocity of the
robot’s joints. It would also be desirable to have a more intuitive and simple method of controlling the robot that does not require the knowledge of any command syntax.

The aim of this project was to upgrade the robot and implement a closed-loop design that would incorporate position feedback, with the aim of meeting the above requirements. Initial investigation revealed that the current design could not be retrofitted with feedback instruments, and so the decision was made to completely overhaul the unit.

Figure 1.2: Home position of the RM-101 (Mitsubishi Electric Corporation 1982)
CHAPTER 1. INTRODUCTION

Specific objectives of the project were to:

- Investigate the feasibility of upgrading the robot to have the capacity for closed-loop feedback control, including the replacement of the DC motors and underlying motor control unit;
- Provide the capacity for future design of a more intuitive human machine interface (HMI);
- Develop the detailed hardware and software design for the upgrade; and
- Refurbish and outfit the robot with the new hardware

1.4 Outline of Thesis

Chapter 2 is a review of literature relevant to the work carried out in this project. As a result of the review, brushless DC motors were selected for use in the project. Consequently, a detailed investigation into the design and topology of brushless DC motor controllers was also carried out. Different feedback control methods and technologies were also investigated and assessed for use in this project.

The end result of the literature review was the design of a brushless DC motor controller to be used in the robot upgrade. Chapter 3 details the process of designing the brushless DC motor controller hardware, and the mechanical modifications required to overhaul the robot, while Chapter 4 discusses the embedded firmware design process and a functional description of the control routines.

Preliminary computer modelling using MATLAB Simulink was carried out and is presented in Chapter 5. Finally, Chapter 6 concludes the thesis. It includes a summary of the practical project achievements and suggestions for further work.
Chapter 2

Literature Review

2.1 Direct Current Motors

2.1.1 Brushed DC Motors

The simplest and most widespread form of direct current motor is the brushed DC motor. The main reason for their popularity is the ability to control their torque and flux easily and independently. A simple DC motor consists of a stator that produces a magnetic field and a rotor which has a large number of copper windings. The windings are often called the armature of the motor. The magnetic excitation field in the stator is produced by either electromagnets or by permanent magnets in the stator.

Torque is developed in a brushed DC motor by current flow in the armature. When voltage is applied to the brushes, a current flows through the windings which induces a magnetic field in the armature. The magnetic field of the stator opposes the induced field in the armature, and thus a powerful torque is produced on the armature causing it to rotate. As the two fields align, the current must be reversed in the armature such that the polarity of the armature field is reversed. The stator and rotor again oppose each other and the motor continues to rotate. This reversal of armature current is performed by a special slip ring known as a commutator. Current is supplied to the commutator through brushes, hence the name of the motor.

As torque is produced by current flow in the armature, and current flow is proportional to the voltage across the armature, then it is obvious that the speed of the motor is proportional to the effective armature voltage. As a coil of wire rotates through a magnetic field a voltage is induced in the windings. This is in actual fact the principle of DC generators, where a prime mover rotates the rotor and a voltage is produced in the armature windings.

In the case of a DC motor, this induced voltage is called the back EMF.
and is given by the Equation:

$$E_b = \frac{Zn\phi}{60}$$  \hspace{1cm} (2.1)

where

- $E_b =$ back EMF [V]
- $Z =$ number of conductors on the armature
- $n =$ speed of rotation [r/min]
- $\phi =$ flux per pole [Wb]

In a permanent magnet DC motor, all of the above terms except for speed of rotation are constants, and are determined by the motor’s construction. This assumption also holds for separately excited DC motors, where the excitation is provided by a separate DC supply and held constant. Rearranging Equation 2.1 then for speed:

$$n = \frac{60E_b}{Z\phi}$$  \hspace{1cm} (2.2)

Grouping the constant terms into an arbitrary electrical constant $K_e$ yields the Equation:

$$n = K_eE_b$$  \hspace{1cm} (2.3)
CHAPTER 2. LITERATURE REVIEW

The constant $K_e$ is called the voltage constant as it relates the motor back EMF to the motor speed; it varies for each motor and can often be found in manufacturer datasheets. The induced back EMF opposes the polarity of the applied source voltage $E_s$, therefore the effective voltage across the armature is given by the equation:

$$E_a = E_s - E_b$$ (2.4)

A DC motor’s back EMF can never rise to equal the supply voltage as the armature voltage would equal zero, and so too would the armature current. There would then be no driving forces on the armature and the motor would stall. However, when a DC motor drives a load between no-load and full-load, the armature voltage is small compared to the supply voltage, and therefore from Equation 2.4 the back EMF is very nearly equal to the supply. (Wildi 2005) Equation 2.3 can then be rewritten as:

$$n \approx K_e E_s$$ (2.5)

This important equation demonstrates the popularity of the brushed DC motor. The output speed is linearly proportional to the voltage supplied to the armature; therefore the speed of a brushed DC motor can be controlled simply by varying the supply voltage. A popular technique for varying the motor speed is to switch the voltage supplied to the motor on and off at a very high frequency. If done fast enough, the armature will only see the average voltage supplied over the time period. Adjusting the ratio of on time to off time varies the average voltage the armature sees, and thus the speed increases or decreases. This method is called pulse width modulation and is a very common technique for controlling many different motor types, as will be seen in later sections.

Positioning systems such as robotic pick and place operations were originally implemented using brushed DC motors due to the relative ease of controlling them. However the high speed repetitive motion of positioning systems subjects the brushes to excessive mechanical wear and sparking, leading to decreased performance over time. The rotor windings must also dissipate their heat through the stator windings as there is no direct path to the outside environment. The extra mass present on the rotor due to the armature windings also decreases the torque to inertia ratio of the motor. (Zribi & Chiasson 1991)

A final side effect of brushed DC motors is the sparking produced by the brushes and the commutation process, which can be a source of ignition and prohibits their use in hazardous environments. Sparking associated with brushed DC motors is also a large source of electromagnetic interference (EMI). (Karnan et al. 2006)
2.1.2 Stepper Motors

The limitations of the brushed DC motor mentioned in Section 2.1.1 have led to the stepper motor’s popularity in positioning applications. Essentially, the stepper motor is a special type of synchronous AC motor. Synchronous motors work on the principle of magnetic attraction. The rotor is magnetically excited (either via permanent magnet construction or by separately exciting coils on the rotor with a DC supply through slip rings) and a rotating magnetic field is created in the stator, usually by a polyphase AC supply. The rotor field and stator field align and thus, as the stator field rotates the rotor is pulled along in sync.

When DC current is injected into a stator coil of a synchronous motor, the magnetic field in the rotor aligns itself with the now static magnetic field in the stator and the motor stalls. In a simple synchronous motor with three stator poles and two rotor poles, the motor shaft may rotate by anywhere up to 60° to align with the energised stator pole. By drastically increasing the number of poles on either the stator or the rotor, this angle can be controlled down to just a fraction of a degree. This is the principle of the stepper motor. In actual fact, the stepper motor is simply a synchronous motor with a high number of salient poles on the rotor and/or stator. Figure 2.2 shows an exploded view of a typical 4-pole stepper motor. The stator poles however have been milled to create a high number of salient poles. Likewise the rotor has also been milled around the circumference for the same reason.

![Exploded view of a hybrid stepper motor](image)

Figure 2.2: Exploded view of a hybrid stepper motor

The high number of poles means that exciting one of the stator coils with direct current results in a very small, discrete movement of the rotor. This property makes stepper motors attractive as the position can be precisely
controlled with DC pulses and no feedback. That is, they can be operated in open-loop with reasonably high accuracy. Knowing the step angle of the motor and the number of steps the rotor has moved though, the output position of the motor shaft can be easily calculated.

Position can also be more finely controlled through half stepping. This is the process of energizing adjacent stator coils so that the rotor will align to a position between the two, allowing the motor to be aligned to half of the actual step angle. With the advancement of modern microcontrollers, microstepping by pulse-width modulation of the stator coils allows even finer precision of the rotor position to just a fraction of the step angle. Consideration must be taken when microstepping however, as it effectively reduces the torque of the motor. This is because PWM essentially reduces the average current applied to the motor, which reduces torque due to the proportional relationship of torque and motor current.

While the inherent stepping ability allows for simple open-loop control, the stepper motor suffers from a number of instabilities. When operated in an open-loop configuration, stepper motors exhibit step responses with significant overshoot and long settling times (see Figure 2.3). At high stepping rates the natural oscillatory response of the motor can also lead to loss of synchronism. This occurs when the overshoot is large enough to cause the motor to move to the next step position, which is a complete rotor tooth pitch from the expected position. (Zribi & Chiasson 1991) The consequence of this high frequency instability is that the motor may misstep, even though the load torque is less than the rated pull-out torque of the motor.

![Figure 2.3: Typical open loop response of a stepper motor (Athani 2007)](image)

Many other researchers have commented on the instability of operating stepper motors in an open loop configuration, and many attempts have been made to understand and control the phenomena. As well as the high-frequency instability mentioned by Zribi & Chiasson above, there is also the mid-frequency instability (also known as mid-frequency resonance or local
instability) that usually occurs at stepping rates lower than 1000 pulses per second. (Cao & Schwartz 1999) Mid-frequency instability has been widely recognised for a long time, though a complete understanding of it has not been well established. Regardless, the instability manifests itself as a range of speeds where pull-out torque is drastically reduced (see Figure 2.4).

![Figure 2.4: Stepper motor mid frequency oscillation (Athani 2007)](image)

Although the properties of stepper motors make them attractive as low-cost open loop positioning devices, they suffer from a number of instabilities and unstable phenomena. Care must be taken when designing the stepper motor controller to ensure that the motor is never operated in its unstable range of frequencies. The overshoot and poor settling times mean that stepper motors are typically only used in low-cost position applications where the cost of feedback devices cannot be justified.

### 2.1.3 Brushless DC Motors

One way to visualise the BLDC motor is as an inverted permanent magnet (PM) brushed DC motor. With a PM brushed DC motor, the excitation is provided by permanent magnets in the stator and the armature windings are on the rotor. With a BLDC motor, typically the rotor is of permanent magnet design and provides the excitation, while the armature windings are on the stator poles. Generally BLDC motors are designed as surface-mount motors, with strong rare-earth permanent magnets placed on the outside of the rotor to provide the excitation field. Currently high performance NdFeB (neodymium-iron-boron) magnets are widely used in BLDC motors. (Krishnan 2001) The motor operates by energising the stator coils, in turn attracting the magnetic rotor and creating torque. In this way, just like the stepper motor mentioned above is a type of synchronous motor, the brushless DC motor is also in actual fact a PM synchronous motor.
It should be apparent from the name that another key difference between brushed and brushless DC motors is the brushes themselves. In a brushed DC motor, current is provided to the commutating ring through brushes. BLDC motors eliminate the brushes and commutating ring arrangement, and instead use separate power electronics to electronically commutate the motor. A small brushless DC motor is shown in the Figure 2.5. The small permanent magnet and windings on the stator poles can be seen.

Typically BLDC motors are three phase machines, therefore requiring a three-phase inverter to commutate the stator windings from a single DC supply. The most efficient operation of a BLDC motor however requires driving only two of the three phases at any one time. One of the main characteristics of the BLDC motor is that the rotor magnets are positioned in such a way that the induced back EMF will be a trapezoidal shape, rather than sinusoidal like a typical PM synchronous motor. To achieve smooth torque production, the stator phase currents are square waves and are applied when the back EMF is constant (see Figure 2.6). (Ozturk 2005)

It can be seen from Figure 2.6 that at any one time there is a constant DC back EMF and constant DC phase current throughout one of the phase pairs. Ideally this results in constant speed and constant torque output for the entire revolution of the motor shaft.

One of the more important factors to consider with brushless motor controller design then is determining when to switch which pair of phases the supply voltage is applied to in order to maintain the constant torque output. Because there is no commutating ring to automatically reverse the magnetic field, the rotor position must be sensed and fed back to the motor control circuit. The motor controller decides which series of
stator coils to energise based upon the current location of the rotor flux vector, such that smooth rotation and torque production is maintained. This is referred to as electronic commutation.

Figure 2.7 shows the commutation states for a typical three-phase BLDC motor, viewed from the driving end perspective. Suppose that phases A and B are energised. Coils A and B will produce two flux vectors, with the resultant stator flux vector located at $\theta_1$. The rotor flux then follows the stator flux to position $\theta_1$. Now suppose that when the rotor reaches $\theta_1$, phases A and C are energised. The resultant stator flux vector is now at position $\theta_2$ and the rotor moves to $\theta_2$. Under forward (counter-clockwise) operation the commutation sequence is AB, AC, BC, BA, CA and CB, which repeats in cyclic order. Reverse (clockwise) rotation is achieved by the commutation sequence CB, CA, BA, BC, AC and AB.
Commutation of a BLDC motor should ideally occur at rotor positions $\theta_1$, $\theta_2$, $\theta_4$, $\theta_5$ and $\theta_6$ to ensure constant torque output. However practically it is difficult to commutate and produce the square wave phase currents at these exact moments. The inductance of the stator windings produces a long electrical time constant that opposes a rapid change in current, therefore slowing down the rise and fall time of the phase currents. For this reason pure square waves cannot be realised, and the torque output drops momentarily in this transient period. In fact, any deviation from the ideal square wave phase currents shown in Figure 2.6 will result in a torque variation. (Hanselman 1994) This is known as torque ripple, and has been the focus of many studies and papers on BLDC motor control.

The major advantage brushless motors have over other competing motor technologies however is the high torque to inertia ratio. Because the armature windings are located in the stator rather than on the rotor, there is less mass on the rotor when compared to a brushed DC motor. This results in less rotor inertia and better torque characteristics of the motor. The lack of windings on the rotor also has many other advantages, such as enabling better cooling and allowing higher voltages to be achieved. (Krishnan 2001) BLDC motors also have the desirable linear voltage-speed relationship that brushed motors have, and can likewise be speed controlled by PWM of the
Further advantages of BLDC motors over other motor technologies include (Ozturk 2005):

- Greater efficiency due to no core losses in the rotor;
- Smaller physical size when compared to brushed DC motors. Recent advances in strong rare-earth magnets such as Samarium–cobalt and NdFeB allow motors with very high flux densities, subsequently allowing high torque motors that are physically smaller and lighter;
- Higher speeds as there are no brush losses to limit the speed of the motor;
- Improved reliability and reduced maintenance and inspection costs due to the lack of brushes;
- No EMI from brush sparking;
- High starting torque; and
- Low noise operation due to removal of brushes and associated brush noise. Commutation is achieved electronically, and the inverter switching frequency is usually high enough so that noise created by harmonics is not audible.

The primary disadvantages of BLDC motors are expense and controller complexity. This is not due to any inherent reason in the motor itself, as a BLDC motor is actually simpler to construct than its brushed counterpart. The higher cost of BLDC motors is due to the fact that a much more elaborate controller is required when compared to brushed DC and stepper motors, along with the need for position sensing elements for correct commutation. The issue of torque ripple can also be a problem if the controller is not well-designed.

If these problems can be overcome however, BLDC motors are an excellent choice for robotic applications due to their desirable torque characteristics, which are well suited to the continual acceleration and deceleration of robotic manipulator joints. For this reason, and the many other advantages listed above, brushless DC motors were selected for this project.

### 2.2 Motor Drive Topologies

#### 2.2.1 Half Bridge

The simplest of DC motor drive topologies is the half-bridge, shown in Figure 2.8. The switches represent transistors that are opened or closed electronically. The resistance and inductance of the stator coils are designated $R_{ph}$.
and $L_{ph}$, and the back EMF of the coils are labelled $e_a$, $e_b$ and $e_c$ for phases A, B and C respectively. When a transistor switch is closed, current is able to flow from $V_{cc}$ through the respective phase and back to the supply.

The parallel diodes are flyback diodes. The purpose of these diodes is to clamp the voltage spike that appears when the voltage supply to an inductive load is suddenly disconnected. In this case, the inductive load is the motor’s stator coil inductance. Without a flyback diode, a large voltage spike would be present when the FET is switched off and the current collapses, which could damage the FET. These diodes are common place in most motor control designs and will appear in following drive topologies as well.

Half-bridges are attractive due to their simplicity and low transistor count, however for brushless motor control the half bridge configuration has many disadvantages. This configuration only allows for positive current flow, and as a result only the positive half cycle of the back EMF waveform shown in Figure 2.6 can be produced for each motor phase. This effectively halves the torque capability of the motor as negative phase currents are not applied. (Hanselman 2006) The limitations of the half bridge also eliminate it from positioning applications such as this project, where high torque is required due to the constant acceleration and deceleration of the robotic arm.

### 2.2.2 H-Bridge

To overcome the limitations of the half bridge topology, a full H-bridge drive can be used. The configuration for a H-bridge is shown in Figure 2.9. This drive topology allows the independent control of each phase current by arranging four transistor switches in a configuration known as a H-bridge.
When an upper transistor (e.g. $S_1$) is on, and a lower transistor at the opposite end of the motor winding (e.g. $S_4$) is turned on, current flows from $V_{cc}$, through the stator winding and back to the supply. If however the opposite pair of transistors are switched on ($S_2$ and $S_3$), current flows in the opposite direction through the windings. The H-bridge configuration therefore allows bipolar current flow and the motor is able to supply torque in both the positive and negative half cycles of back EMF.

![H-bridge drive topology](image)

**Figure 2.9: H-bridge drive topology (Hanselman 2006)**

The disadvantage of the full H-bridge configuration is the complexity of the circuitry. While the phase currents can be individually controlled, a total of twelve transistors are required for a three phase motor. For mass production, the main aspect that prohibits the H-bridge from most applications is the cost of the circuit. (Hanselman 2006) For this project however, the H-bridge topology is not an available option as most small BLDC motors do not provide access to the neutral end of the phase windings. Most commercially available BLDC motors are internally connected in either a wye or
delta topology (discussed further on), therefore isolation and independent control of each phase is not possible without modifications to the internal motor connections.

2.2.3 Wye-Connection

The most common brushless motor configuration is the wye-connection topology shown in Figure 2.10. In this configuration the three motor phases are connected together in the common three-phase wye (or star) configuration. Current flow through the motor is controlled by turning on one of the upper transistors (e.g. $S_1$) and one of the lower transistors at the opposite end of the motor winding (e.g. $S_4$). In this example, current would flow from $V_{cc}$, though windings A and B, and back to the source. Current can be made to flow in the opposite direction through the same pair of windings by turning on transistors $S_3$ and $S_2$. The reason for the wye-connection’s popularity is that bipolar current flow is supported, while requiring half as many transistors as the full H-bridge configuration.

![Figure 2.10: Y-connection topology (Hanselman 2006)](image)

An important factor to note with the wye-connection topology is that the centre neutral point is not connected external to the motor. Therefore two
phases must be energised at any one time to complete the circuit. Because of this, individual control of each phase cannot be achieved like it can in the H-bridge configuration. Another way of realising this limitation is by analysing Kirchoff’s current law, which states the sum of the three phase currents at the neutral point must be equal to zero. The current in the two active phases will be in phase, therefore the third winding must be equal to the negative sum of the two. (Hanselman 2006) The result is that all three-phases are mutually dependent on one another and cannot be independently controlled.

2.2.4 Delta-Connection

An alternative to the wye-connection mentioned earlier is the delta-connection topology, shown in Figure 2.11. With this configuration, the motor phases are connected to form a loop (or delta). Current flow through the motor is again created by switching on an upper and lower transistor pair, as with the wye-connection and H-bridge configurations. Delta-connections typically provide higher motor speed than wye-connected motors though produce less torque, by a factor of $\sqrt{3}$. Inversely, wye-connected motors produce more torque and a lower top speed by the same factor.

Figure 2.11: Delta-connection topology (Hanselman 2006)

In a wye-connected load the three phase currents must sum to zero at all times, while the voltages are unconstrained. A delta-connected motor is the opposite of this. The voltages must sum to zero at all times while the currents are unconstrained. If the voltages are not balanced in a delta connected load, a current will circulate within the delta and cause additional
voltage drop across the load. In the case of a motor, this results in additional ohmic losses that do not produce any additional torque.

With delta-connected motors, the criteria for circulating currents is satisfied when triple-n harmonics (where \( n = 1, 2, 3, \) etc) are present in the back EMF. These triple-n harmonics will be in-phase, therefore summing and creating a voltage imbalance. Unfortunately, Fourier series expansion of the trapezoidal back EMF produced by BLDC motors shows a significant third order harmonic, therefore circulating currents will be produced when a BLDC motor is delta-connected. These currents result in additional braking torque on the motor and increased torque ripple. (Elliott & Bowling 2004) A solution to the issue of circulating currents in delta-connected motors is to use a BLDC motor with a sinusoidal back EMF shape. This is because sinusoidal back EMFs do not contain third order harmonics, therefore circulating currents will not be produced in the delta. However, the torque ripple of a delta-connected sinusoidal motor is still greater than that of wye-connected trapezoidal one.

Despite the immediate disadvantages of delta-connected motors, it may actually be advantageous to use a delta connection at higher speeds. This is because the back EMF of a delta-connected motor will be less than that of a wye-connected one, therefore greater power is available to the motor shaft.

2.3 Position Feedback

2.3.1 Hall Effect Sensors

The Hall Effect was discovered in 1879 by Edwin Hall, who found that when a current-carrying conductor is placed inside a perpendicular magnetic field, a voltage will appear across the conductor that is perpendicular to both the current and the magnetic field. When a perpendicular magnetic field is present, a Lorentz force will act upon the current causing a charge build-up at the bottom of the conductor and producing a measurable voltage. (Wilson 2004) This voltage is termed the Hall voltage, and is the basis of the Hall Effect sensor. BLDC motors make use of the Hall Effect by embedding Hall sensors in the stator of the motor. As the rotating magnetic poles of the rotor approach the sensor, a Hall voltage is developed across the element. This indicates the rotor flux vector is nearby and the Hall sensor outputs a logic high signal.

A common configuration with BLDC motors is to have three Hall sensors spaced 120° around the circumference of the stator. The resolution of the Hall sensors is typically 180°—that is, a single Hall sensor will produce a logic high signal when the rotor flux is within 180° of the sensor, and a low signal for the remaining 180°. By spacing three sensors around the stator, the combined resolution can be increased to 60°. By inspection of Figure
2.13, it can be seen that the three Hall sensors output a unique binary code every $60^\circ$. Fortunately this resolution is fine enough for commutation purposes, and it is common to use the Hall sensors as the input to the inverter switching logic. Using Figure 2.13 as a reference, the following commutation logic can be obtained. Table 2.1 determines which phase pair to selected based upon the current Hall sensor output.

<table>
<thead>
<tr>
<th>$H_a$</th>
<th>$H_b$</th>
<th>$H_c$</th>
<th>Commutation state</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>CA</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>CB</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>AB</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>AC</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>BC</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>BA</td>
</tr>
</tbody>
</table>

Table 2.1: Brushless DC commutation logic

A major advantage of Hall sensors is that they also ensure that the motor can start-up smoothly. Due to the PM construction of the rotor, a rotor flux vector will be present even when the motor is not rotating; thus the approximate rotor position is available even when the motor is stationary. Without any position information an alignment stage would be needed, which could result in back rotation of the motor shaft. These advantages make Hall sensors useful for BLDC motor commutation, though for precise positioning an additional position feedback device with finer resolution would be required. While the Hall sensors will indicate in which $60^\circ$ segment the rotor is aligned, it cannot provide the absolute position of the rotor within that $60^\circ$ arc. This uncertainty would result in a large position error, making them unsuitable for positioning the end effector of a robotic manipulator.
Figure 2.13: Relationship between BLDC phase currents and Hall sensor outputs
2.3.2 Incremental Encoders

Rotating incremental encoders are very popular and relatively inexpensive devices for position feedback. Typically these are optical devices which employ a slotted encoder wheel. A light emitter, such as an LED, is positioned on one side of the encoder wheel and a photodiode on the other. The encoder wheel is coupled to the motor shaft such that when the motor turns, the light from the emitter either passes through a slot in the encoder wheel or is interrupted by the wheel itself. This produces a series of pulses at the photodiode output, and the relationship between these pulses can be used to determine the position and speed of rotation of the motor.

Quadrature encoders employ two slotted tracks on the encoder wheel rather than one, as well as a second photoemitter-detector pair. The two tracks are slightly offset from one another so that the photodiode outputs will be 90° out of phase electrically (see Figure 2.14). This is where the encoder derives its ‘quadrature’ namesake from. Encoder technical specifications usually state the number of pulses (or lines) per revolution. Considering that the two pulses are 90° phase shifted with one another, there are four possible logic states for the encoder output. Since each full cycle contains four binary transitions, an encoder with a specification of 1000 pulses per revolution will actually generate 4000 2-bit binary transitions for one full rotation of the encoder wheel.

![Figure 2.14: Quadrature outputs showing phase shift and binary transitions](image)

When decoding the quadrature output, these binary transitions are detected and used to clock a high speed counter. The relationship between the counts provides the speed and position feedback information, while the sequence of transitions determines the direction of rotation. That is, if channel A leads channel B, then it is known the motor is rotating clockwise; otherwise, if channel B leads channel A the motor is rotating counter-clockwise.
Decoding the quadrature output was originally based upon edge detection. A transition from high to low (or vice-versa) on one of the encoder’s outputs would trigger the counter to count up or down, depending on the current state of the second output. Due to advancements in modern electronics and microcontrollers, modern quadrature decoders instead use a high speed clock to sample the state of the two outputs. When a change is detected the decoder counts up or down, depending on the previous value (see Table 2.2).

<table>
<thead>
<tr>
<th>Forward (count up)</th>
<th>Reverse (count down)</th>
</tr>
</thead>
<tbody>
<tr>
<td>From</td>
<td>To</td>
</tr>
<tr>
<td>01</td>
<td>11</td>
</tr>
<tr>
<td>11</td>
<td>10</td>
</tr>
<tr>
<td>10</td>
<td>00</td>
</tr>
<tr>
<td>00</td>
<td>01</td>
</tr>
</tbody>
</table>

Table 2.2: Quadrature decoding sequence

Knowing the resolution of the encoder (degrees per count), the angular position of the rotor can easily be determined by multiplying the counter output by the resolution. For example, knowing that the encoder produces 4000 counts per revolution, the resolution would be \( \frac{360^\circ}{4000 \text{ counts}} \), or 0.09° per count. If the counter output is 100, then the motor has rotated 9°. Rotational speed is also easily determined from the quadrature output by measuring the time between counts with a high speed timer. The encoder resolution indicates how many degrees of rotation occur between counts, therefore dividing the encoder resolution by the time between each count gives an accurate indication of the motor’s rotational speed.

The disadvantage of quadrature encoders is that they do not immediately provide absolute position information. The position determined from decoding the quadrature output is only relative to the previous position, therefore the starting coordinates of the device must be known to accurately determine the absolute position of the machine. However depending on the situation, this limitation can be circumvented by the addition of a third signal called the index. This involves the addition of a third track to the encoder disk with just a single slot, resulting in just one output pulse per revolution. Knowing the rotor angle at this index position means the absolute position can be determined. For example, consider that all but the wrist joints of the Movemaster RM-101 cannot move through an entire 360° rotation (see Section 1.2.1). If the motor shaft is directly coupled to the output, then the index signal will appear just once between the angular limits of each joint. The index signal is therefore an indication of one particular angular position for each joint, and can be used as an absolute position reference. The use of a three channel quadrature encoder with an
index output would therefore be beneficial for this particular application.

2.3.3 Absolute Encoders

Absolute encoders are typically used for rotational applications where the absolute position of the device must be known when the reference information is lost (such as upon power loss). They consist of an encoder wheel that has multiple slotted tracks with a unique pattern and single detector for each track. In this way, each quantised position in a single revolution will produce a unique binary number. As each number corresponds to a unique position, simply reading the detector outputs at startup provides the current absolute position of the device.

The encoder tracks are usually produced in such a way that as the encoder wheel rotates, the detector output increments in either a natural binary or Gray coded sequence. Of the two, Gray coded encoders are much more popular than natural binary coded encoders. The reason for this is the elimination of ambiguities in component tolerances. For example, when going from position 255 to position 0 in Figure 2.15b, all eight bits must toggle from 1 to 0. There is no certainty that the detector outputs will toggle at the exact same instant, therefore there is considerable ambiguity during a state transition. (Borenstein, Everett & Feng 1996) Because of the asynchronous switching times of the detector outputs, there may be many different signals that appear at the controller input as all the bits toggle state. The controller will register this as a series of sporadic position changes over a very short time. As only one bit changes between each transition in the Gray code, this ambiguity is eliminated as only one detector output has to be toggled at any one time.

Figure 2.15: Gray coded (a) and binary coded (b) absolute encoder wheels (Borenstein, Everett & Feng 1996)

While absolute encoders are useful for providing a high resolution abso-
lute position, they have a number of potential disadvantages. It should be obvious that absolute encoders are larger than incremental encoders as more tracks must be placed on the wheel. This increase in size decreases shock and vibration tolerances, as well as increases cost. In fact, a general rule of thumb is that while each additional encoder track doubles the resolution, it quadruples the cost. (Avolio 1993)

Finally, the parallel data output of an absolute encoder increases the number of wires and electrical connections. A 13-bit absolute encoder that uses complimentary output signals to improve noise immunity would require 28 conductors, compared to just six for a quadrature encoder. (Borenstein, Everett & Feng 1996)

2.3.4 Back EMF Detection

When a BLDC motor is operating only two phases will be energised at any one time, therefore providing an opportunity to measure the back EMF induced across the undriven phase. As this back EMF varies over time, it provides a clever analogue representation of the rotor flux position.

Sensorless commutation of BLDC motors relies on the back EMF measurements and zero crossing detection (the point where the back EMF changes from positive to negative, or vice-versa). The main problem with sensorless commutation is that the rotor is in a completely unknown position when the motor is at rest. Using Hall sensors the approximate position of the rotor is always available, within an accuracy of 60 electrical degrees, which is enough to select the appropriate phase pair to ensure a smooth startup. However, as back EMF is directly proportional to speed, when the motor is at rest there is zero back EMF and thus no available position information.

Typically, sensorless motors will use an alignment stage to get around this problem. (Gambetta 2006) This requires a predefined first commutation state which is chosen arbitrarily. The exact commutation state is not important, as each state corresponds to a known rotor position. For example, referring to Figure 2.7, if alignment is carried out on state AB then the rotor will align to position $\theta_1$. However if BC is chosen as the starting commutation state, the rotor will align to position $\theta_3$. It should be obvious then that the chosen starting state does not matter, as any commutation state corresponds to a known position, and correct commutation can continue from there. This alignment stage however may cause back rotation of the rotor. For example, if the rotor is initially located at $239^\circ$ and AB was chosen for the alignment stage, the motor would align itself to $\theta_1$ by backwards rotating $179^\circ$.

Gambetta also notes that the back EMF method suffers from a number of other limitations. At low speeds the back EMF induced will be excessively low, therefore making it impossible to detect the correct commutation po-
CHAPTER 2. LITERATURE REVIEW

sitions. For this reason the system must run in an open-loop configuration at lower speeds (typically 5 percent of rated speed). Startup is likely to be rough because, without any position feedback, estimations must be made about when to commutate each state. Actual rotor position at any time depends on a number of variables such as speed, moment of inertia for the motor–load combination, non-inertial loads such as friction, and electrical quantities such as voltage and current; therefore these estimations can be extremely difficult to make and smooth startup cannot be guaranteed.

The potentially rough startup and back rotation, coupled with the open loop operation at low speeds are obviously undesirable for precise positioning applications such as robotics. The back EMF technique is however very economical for applications that do not require a smooth startup and undergo continual rotation, such as fans and pumps.

2.4 BLDC Controller Design

Generally speaking, a BLDC motor controller consists of three parts: the motor itself, a power converter, and a speed, torque and current controller (such as a microcontroller). As mentioned previously in Section 2.1.3, a brushless DC motor is a type of permanent magnet synchronous motor that is driven by DC phase currents. In actual fact, the BLDC motor is a polyphase AC machine, however it is desirable to operate them from just a single DC supply. Therefore BLDC motors are usually driven by a three-phase inverter which is constructed from three half-bridge networks.

A simplified BLDC controller is shown in Figure 2.16, where power electronics are used to construct the three-phase inverter. Here power MOSFETs (Metal-Oxide-Semiconductor Field Effect Transistor) are shown as the switching elements. Newer IGBTs (Insulated-Gate Bipolar Transistor) are becoming more popular due to their higher collector current capabilities, therefore allowing for higher power motor drives. However the power MOSFET has faster switching characteristics and is still the best choice for low-voltage, high frequency applications such as small BLDC motor control. (Bose 2006) Flyback diodes are also shown across the MOSFETs to clamp the inductive voltage spike that would appear when a motor winding is disconnected. Rotor position is sensed by Hall sensors embedded around the motor’s stator circumference. The Hall sensors output a 3-bit binary signal to the microcontroller that indicates the rotor’s current location. From this the microcontroller then decides which MOSFET pair to activate so that the correct commutation state is selected and the motor continues rotation.

While Figure 2.16 provides a reasonable circuit for driving a BLDC motor in continual rotation, more accurate instrumentation is required for precise positioning applications such as robotic pick-and-place operations. For this project, precise position and speed feedback is taken from a quadrature
encoder mounted on the motor shaft. The quadrature encoder provides position feedback accurate to within a fraction of degree, making it useful for fine positioning of the rotor. The microcontroller also determines the velocity information from the encoder outputs, and controls the motor speed. Speed control is usually implemented with a PWM type scheme to vary the average current supplied to the motor windings, therefore reducing motor speed and torque. The PWM duty cycle is adjusted by the microcontroller until the motor speed matches a user setpoint.

The circuit shown in Figure 2.16 is a common implementation for a BLDC motor controller. Some variations could include using back EMF detection to sense the rotor position and/or a full H-bridge inverter, though these both have their disadvantages such as poor performance at low speeds and increased circuit complexity.

### 2.4.1 Inverter Switching Schemes

Speed control of a BLDC motor is usually controlled via pulse width modulation, where the supply voltage is chopped at a fixed frequency with a duty cycle depending on the current error. With this technique both the current and rate of change of current can be controlled. One of the primary advantages of PWM speed control is that the switching frequency is a fixed parameter, and therefore acoustic and electromagnetic noises are relatively easy to filter. (Akin & Bhardwaj 2010)

A study by Wiberg (2003) showed that PWM frequency is not vital to the performance of the motor. In this study it was found that tests with PWM
frequencies between 1kHz and 15kHz showed good results, and performance is not drastically affected until the PWM frequency exceeds 25kHz. For this project, 18kHz was selected so that the switching frequency is at the edge of the human audible range.

Figure 2.17: Inverter hard chopping

Chopping of the supply voltage is handled by power transistors such as MOSFETs or IGBTs, which conduct or interrupt the supply voltage depending on the state of the gate input. The high frequency switching of the transistor gate input is usually provided by a microcontroller, and two different switching techniques are commonly used. These are hard chopping and soft chopping. Regardless of which method is selected, both will produce the same basic low frequency envelope (average conducted current) to the motor.

In the hard chopping technique, both the high and low side transistors are modulated at the same time by identical PWM signals (see Figure 2.17). The advantage of this technique is that the controller is much easier to design and is also less expensive as it only needs to handle three PWM signals. Since switching is symmetrical, power losses are also equally distributed between transistor switches. Transistor conduction losses can also be ignored as no transistor is continually conducting for any 120° period. Power loss
of the inverter is instead strongly determined by just the duty cycle and switching frequency. (Keskar et al. 2005) The disadvantages of this technique however is that current ripple, EMI and acoustic noise are all increased when compared to soft chopping techniques.

The soft chopping approach involves leaving one transistor on while switching the other. Usually the high side transistor modulates the duty cycle, while the low side transistor steers the current continuously for 120°. The soft chopping technique also has a variant loosely referred to as 60° chopping. In this case, both the high and low side transistors are modulated. The upper transistor is switched for 60° and operates in continuous conduction for the remaining 60°. Inversely, the low side transistor is continuously conducting while the high side transistor is switching, and is switched while the high side transistor is left on. At any one time, only one switch is switching while the other is in conduction.

![Figure 2.18: Inverter soft chopping](image)

Whether the high side or low side transistor is switching depends upon the state of the undriven phase. When the back EMF of the undriven phase is positive, the high side transistor is switched, while the low side transistor is switched when this voltage is negative. Both soft chopping variations provide the same low frequency envelope. The difference between the two
variations is that switching and conduction losses are shared between the high and low side transistors when operated in 60° switching. (Keskar et al. 2005)

Soft chopping minimises inverter switching losses as only one transistor is switched at any one time, though conduction losses must also be considered for a full representation of the inverter soft-switching losses. This technique also produces lower torque and current ripple when compared to hard chopping, as well as lower acoustic noise. The switching associated with power electronics under PWM has also been identified as a large source of EMI. The soft chopping technique therefore also reduces EMI when compared to hard chopping as only one transistor is switched at any one time. The main disadvantage of soft chopping however is the increased controller complexity, as six independent PWM signals must be generated. (Akin & Bhardwaj 2010)

2.4.2 MOSFET Characteristics

As stated earlier in Section 2.4.1, the inverter stage of most BLDC controllers is typically implemented using power MOSFETs. In Section 2.2 these MOSFETs were shown as simple ideal switches that are activated
electronically. However due to the construction of the devices and their arrangement within the inverter topology, consideration must be given to the actual circuitry used to activate the MOSFETs.

The MOSFET is a three-terminal device, consisting of a gate electrode that is insulated from a semiconductor substrate and the other two contacts (the drain and the source). Applying a voltage to the gate induces a conductive channel in the semiconductor substrate that connects the drain and source terminals, permitting current to flow from one to the other. The insulation however forms an internal capacitance that must be charged and overcome before current can begin to flow from the drain to the source. The voltage required to reach this point is called the threshold voltage, and is usually designated as $V_{gs(th)}$. Figure 2.20 shows the cross-section of an n-type MOSFET showing the inter-junction capacitances.

When using the MOSFET as a high frequency switch, the device driving the gate terminal of the MOSFET should be able to supply a large current very quickly to minimise the time spent charging the input capacitance and the time spent in the linear region of the MOSFET’s operation. (Pathak 2001) Furthermore, to fully enhance a standard MOSFET, a gate to source voltage of around 10V is usually required. Typically the outputs of most microcontrollers do not meet these requirements, therefore making them ineffective at driving MOSFETs directly. For these reasons a dedicated MOSFET driver should be placed between the microcontroller outputs and the MOSFET inverter circuitry. These devices take the logic output from the microcontroller and shift it to a level suitable for driving the MOSFETs.

To determine the current requirements of the MOSFET driver, Equation 2.6 should be used:

$$I = \frac{Q}{dt} \quad (2.6)$$
CHAPTER 2. LITERATURE REVIEW

where

\[ I = \text{charging current [A]} \]
\[ Q = \text{MOSFET total gate charge [C]} \]
\[ dt = \text{charging time [s]} \]

Equation 2.6 indicates that the more current the MOSFET driver can supply, the faster the gate capacitance can be charged and the faster the switching will occur.

Three-phase motor inverter circuits require a high-side and low-side switch arranged in a half-bridge configuration for each phase, as shown in Figure 2.16. The simplest method to construct this circuit is to use n-type MOSFETs for the high-side switches and p-type MOSFETs for the low-side. However, the issue with this approach is that it requires two voltage supplies—a positive supply for the high-side and a negative supply for the low-side.

Using all n-type MOSFETs for the configuration in Figure 2.16 would result in the high side switches not operating. This is because the high-side MOSFET source potential is not referenced to ground—it floats at the potential of the low-side drain, \( V_d \). With respect to ground, \( V_{gs} \) for the high-side MOSFET must be greater than the sum of \( V_d \)(low) and \( V_{gs(th)} \)(high) for conduction to begin. In actual fact, if a standard MOSFET is to be used as the high-side switch, \( V_{gs} \) would have to be greater than \( V_d \) of the low-side MOSFET by at least 10V for full enhancement and efficient operation of the switch.

The result of this realisation is that if a single voltage supply is used to drive both the high-side and low-side switches, the voltage must rise above the common voltage rail to activate the high-side devices. This can be achieved by using a bootstrap circuit as shown in Figure 2.21 to lift the voltage seen by the high-side switches. In this circuit, when the low-side switch is turned on, its \( V_d \) ideally drops to ground potential. This allows the bootstrap capacitor to charge to \( V_{cc} \). When the low-side switch is turned off again, \( V_d \) suddenly rises, however the bootstrap capacitor still maintains a potential difference of \( V_{cc} \) across its terminals. Therefore, with respect to ground, the capacitor’s voltage is now equal to \( V_{cc} + V_d \). This bootstrap voltage can then be used to drive the gate terminal of the high-side device.

There are a number of considerations to take into account when selecting the size of the bootstrap capacitor. The bootstrap capacitor provides the charge required to turn on the high-side switch, and therefore discharges when the switch is on. The capacitor must be large enough so that the bootstrap voltage does not drop below the threshold voltage required to keep the switch turned on. The maximum allowable capacitor voltage drop
Figure 2.21: Basic bootstrap power supply circuit

$(\Delta V_{BOOT})$ is determined by the equation:

$$\Delta V_{BOOT} = V_{cc} - V_f - V_{gs(th)}$$  \hspace{1cm} (2.7)

where

$V_{cc}$ = Gate driver supply voltage [V]

$V_f$ = Bootstrap diode forward voltage drop [V]

$V_{gs(th)}$ = Switch gate-to-source threshold voltage [V]

The value of the bootstrap capacitor is a function of the total amount of charge supplied by the capacitor while the switch is conducting, and the maximum allowable voltage drop over the same period. The total charge supplied by the bootstrap capacitor is calculated by:

$$Q_{TOTAL} = Q_{GATE} + (I_{LKCAP} + I_{LKGS} + I_{QBS} + I_{LK} + I_{LKDIODE}) \cdot t_{ON} + Q_{LS}$$  \hspace{1cm} (2.8)
CHAPTER 2. LITERATURE REVIEW

where

\[ Q_{GATE} = \text{Total gate charge [C]} \]
\[ I_{LKGS} = \text{Switch gate-source leakage current [A]} \]
\[ I_{LKCAP} = \text{Bootstrap capacitor leakage current [A]} \]
\[ I_{QBS} = \text{Gate driver quiescent current [A]} \]
\[ I_{LK} = \text{Gate driver leakage current [A]} \]
\[ I_{LKDIODE} = \text{Bootstrap diode quiescent current [A]} \]
\[ t_{ON} = \text{High-side switch on time [s]} \]
\[ Q_{LS} = \text{Charge required by MOSFET driver’s internal level shifter [C]} \]

The capacitor leakage current is only important if an electrolytic capacitor is used, and can be ignored otherwise. Knowing that \( C = \frac{Q}{V} \), the absolute minimum value of the bootstrap capacitor is then determined by the equation:

\[
C_{BOOT} = \frac{Q_{TOTAL}}{\Delta V_{BOOT}} \]
\[
= \frac{Q_{GATE} + (I_{LKGS} + I_{QBS} + I_{LK} + I_{LKDIODE}) \cdot t_{ON} + Q_{LS}}{V_{cc} - V_f - V_{gs(th)}}
\]

The other element to consider is the bootstrap diode. When the bootstrap voltage exceeds \( V_{cc} \), the diode becomes reverse-biased and prevents the capacitor from feeding back into the voltage supply. However practically a diode can not switch from a conducting to non-conducting state instantaneously—the time it takes to do this is called the reverse recovery time. The bootstrap diode should be a fast or ultra-fast recovery diode to limit the amount of charge fed back to the voltage source and minimise leakage current. The bootstrap diode also needs to be able to block the full power rail (the voltage applied to the motor), which is applied to the diode cathode terminal when the high side switch is conducting.

2.4.3 Overload Protection

Electric motors should be protected against overload conditions, where the motor is drawing more current than it is rated for. Overload protection is achieved with a current sensing element, of which the most common is the shunt resistor. This method involves placing a small resistance in series with the load and measuring the voltage across the resistor. Knowing the resistance value and the voltage produced across the element, the current through the resistor can be easily calculated. By conservation of electric charge, the current flowing through the motor must be the same. Therefore
the voltage produced across the resistor is a direct indication of the current flowing through the motor. Knowing the full-load current of the motor, the expected voltage across the resistor can be calculated. This value can be used as a tripping setpoint; that is, if the measured voltage rises above this level, then the motor must be drawing more than its rated full load current and is operating in an overload situation.

The disadvantage of using a shunt resistance is that it is an intrusive method of current sensing. The resistor must be placed in series with the load circuit to be measured, and therefore the total voltage available to the load is the difference between the supply voltage and the voltage drop across the resistor. For this reason, shunt resistors are typically very small (in the range of 10-15 milliohms), such that the effect on the load voltage is negligible. It is for this reason also that an amplifying measurement circuit must be used. Typically a non-inverting op-amp circuit is used to amplify the sensed voltage to a level suitable for measurement and comparison (by a microcontroller for example).

![High-Side Current Measurement](image1)

![Low-Side Current Measurement](image2)

Figure 2.22: High and low-side resistive shunt measurement (Lepkowski 2003)

Shunt resistors may be used as either a high-side or low-side measurement device. A high-side monitor places the resistor in series with the power source and the load, while a low-side monitor places the resistor between the load and ground. High-side measurement is usually preferred, but has the disadvantage that high voltage components must be used as the sensed voltage will be very close to the supply voltage. For example, consider a 24V power source supplying 1A to the load through a shunt resistor of 10mΩ. The voltage drop across the resistor in this case would be just 10mV, meaning the measurement circuit in Figure 2.22 will see 24V on its non-inverting terminal and 23.99V on the inverting terminal. Therefore expensive high-voltage op-amps must be used in the high-side measurement circuit.

Low-side measurement provides the advantage that the measured voltage is referenced to ground, therefore standard low-voltage op-amps can be used. However, low-side measurement has a number of issues. The shunt resistor disrupts the ground path, producing an offset voltage which can cause EMI
and noise problems. The effect of EMI on a measurement can result in a large DC offset at the output of the op-amp, giving an unreliable measurement. Also the measured voltage is likely to be only a few millivolts above the ground potential, which is used as the lower voltage rail in the op-amp circuit. Op-amps typically exhibit poor linearity around their voltage rails, also affecting the reliability of the measurement. For these reasons, low-side measurement circuits should be combined with an EMI filter and an offset circuit to lift the shunt voltage to around the middle of the op-amp’s operating range. This results in the relatively complex measurement circuit shown in Figure 2.23.

A final disadvantage of low-side measurement is that it can not detect a fault where the load is accidentally connected to ground via an alternate path. If the motor is abnormally grounded and the measurement circuit is bypassed, the controller will not know and will simply assume the motor is drawing no current at all.

An alternative to the shunt resistor method is to use an isolated sensing element, of which two popular choices are Hall Effect current sensors and current transformers. The Hall Effect current sensor works in a similar manner to the position sensors described in Section 2.3.1. As current flows through a conductor a magnetic field is produced. This produces a Hall voltage in the sensing element, which gives an analogue representation of the load current. Hall Effect sensors are popular as the sensing element is isolated from the load circuit, minimising the effect of the measurement circuit on the load. They are also available in prefabricated ICs containing the internal EMI filtering and amplification, therefore drastically simplifying the measurement circuit. The isolation property of the Hall Effect sensor also means it can be used as either a high or low-side measurement device.

Current transformers operate on the principle that current flowing in
The primary winding will induce a current in the secondary winding that is proportional to the turns ratio of the two sides. The most common design for a current transformer is a coil of wire wrapped around a steel ring, which is passed over the current-carrying conductor to be measured. In this way, the transformer has a primary winding consisting of one ‘turn’ (the conductor to be measured itself) and a secondary winding of many tens to hundreds of turns. Therefore a small current suitable for measurement will be induced on the secondary side that is proportional to the load current. Current transformers have the advantage of complete galvanic isolation, allowing very high currents to be measured, however only AC measurements can be made as DC current would saturate the transformer.

<table>
<thead>
<tr>
<th>Current Sensing Method</th>
<th>Shunt Resistor</th>
<th>Hall Effect</th>
<th>Current Transformer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>Good</td>
<td>Good</td>
<td>Medium</td>
</tr>
<tr>
<td>Accuracy vs. Temperature</td>
<td>Good</td>
<td>Poor</td>
<td>Good</td>
</tr>
<tr>
<td>Cost</td>
<td>Low</td>
<td>High</td>
<td>Medium</td>
</tr>
<tr>
<td>Isolation</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>High Current Measuring Capability</td>
<td>Poor</td>
<td>Good</td>
<td>Good</td>
</tr>
<tr>
<td>DC Offset Problem</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Saturation/Hysteresis Problem</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Power Consumption</td>
<td>High</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td>Intrusive Measurement</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>AC/DC Measurements</td>
<td>Both</td>
<td>Both</td>
<td>Only AC</td>
</tr>
</tbody>
</table>

Table 2.3: Comparison of current sensing methods (Lepkowski 2003)

2.5 Feedback Control

The principle of closed-loop feedback control is relatively simple. A transducer, or sensor, measures the output of the system and produces a proportional signal. The output signal is fed back to the input, where it is compared to the reference input value—which is the user’s desired output of the system. The difference between the reference signal and the physical output is the error signal, which is sent to the controller for processing. The goal of the controller is to progressively adjust the input parameters of the
plant to drive the error signal to zero. The most obvious case for feedback control in this project would be to minimise the position error of the robotic manipulator, however some other uses for feedback control could be for speed control, torque control or implementing complex movement profiles.

Figure 2.24: Block diagram of a basic feedback loop (Åström 2002)

2.5.1 PID Control Theory

The familiar PID controller is the most common form of feedback controller, and has been used in some form for over a century. In process control today, more than 95% of control loops are implemented with PID controllers, though the majority operate in just PI mode. (Åström 2002) Essentially the PID controller calculates three values depending on the process error. These three values are the proportional error, integral error and derivative error, and the weighted sum of these is used to adjust the system.

The proportional term acts on the current error, by multiplying the process error with a constant gain. Therefore as the gain increases, the proportional output of the controller will increase to compensate.

\[ P_{out} = K_p e(t) \] (2.11)

A purely proportional controller will never settle at the target value and will always have a steady state-error. This is because the output is directly related to the magnitude of the error signal. As the error signal approaches zero, so does the output of the controller. Therefore an error is required to maintain an output from the controller. Increasing the proportional gain \( K_p \) will decrease the steady-state error, but will make the system more sensitive to changes in the output and will tend towards a system that oscillates. (Åström 2002)

The integral error acts on the sum of the past errors. If the total amount of process error is increasing, the integral term will increase to compensate.

\[ I_{out} = K_i \int_0^t e(\tau)d\tau \] (2.12)
A large problem with integral control is the issue of integral wind-up. One common scenario that can cause integral wind-up is control system override, where another controller takes over control of the system (e.g. for safety reasons). If the original controller is not switched off it will continue to receive an error signal that will accumulate over time due to the integral term. The simplest solution to prevent integral wind-up is to notify the controller of the override condition, and using logic within the controller to break the control loop if the plant is being overridden. This method is also known as external reset feedback. (Willis 1999) Integral wind-up may also occur due to saturation of the system, or if the setpoint is set outside the physical limits of the plant. In these instances, the controller will not be able to drive the system to zero error, causing the integral error to wind-up over time. In these cases, more advanced anti-windup techniques are used such as back-calculation and tracking of the integral term. (Åström 2002)

Finally the derivative term anticipates future error by calculating the rate of change of the process.

\[ D_{out} = K_d \frac{d}{dt} e(t) \]  

(2.13)

The derivative term improves the dynamic response of the control loop, though it can cause instability if noisy signals are present. Differentiation of a signal will amplify noise, therefore making the system highly sensitive to fluctuations in the error signal due to noise. If the derivative gain is large enough, noise in the error signal can cause the system to become unstable.

Summing the result of the proportional, integral and derivative actions gives the textbook PID controller. Mathematically the output of the controller is given by the equation:

\[ u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{d}{dt} e(t) \]  

(2.14)

The process of tuning a PID controller involves finding a balance between the proportional, integral and derivative gain values that gives the best compromise between overshoot, rise time and settling time for the given system to be controlled.

**Ziegler-Nichols Method**

To avoid the complexities of traditional modelling and analysis, an aggressive method for PID tuning was developed by John G. Ziegler and Nathaniel B. Nichols in the early 1940s. This method is now a very common method for simple loop tuning and is known as the Ziegler-Nichols method (or ZN method for short).

Essentially the ZN method is as follows. The integral and derivative gains are first set to zero. The proportional gain is then increased until
it reaches the ultimate gain, $K_u$. The ultimate gain is the point at which the output of the loop oscillates at constant amplitude. The period of the oscillation, $T_u$, and the ultimate gain $K_u$ are used for the calculation of $K_p$, $K_i$ and $K_d$ as follows (Ziegler & Nichols 1942):

<table>
<thead>
<tr>
<th>Control type</th>
<th>$K_p$</th>
<th>$K_i$</th>
<th>$K_d$</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>$0.5K_u$</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>PI</td>
<td>$0.45K_u$</td>
<td>$1.2K_p/T_u$</td>
<td>–</td>
</tr>
<tr>
<td>PID</td>
<td>$0.6K_u$</td>
<td>$2K_p/T_u$</td>
<td>$K_pT_u/8$</td>
</tr>
</tbody>
</table>

Table 2.4: Controller gains as per the Zeigler-Nichols tuning method

The potential disadvantage of the ZN method is that it is a form of online tuning, which means that the system or process is operational while the loop tuning is being performed. This can upset the process and requires some trial-and-error to implement.

### 2.5.2 In Relation to Servo Motors

Typically BLDC and other servo motors use multiple cascaded feedback controllers for complete control of the motor. For example, the output position is measured with a position sensor such as an encoder. Differentiating the position will provide information on the speed, allowing a speed controller to be placed inside the position control loop. The only requirement of cascading controllers is that the time constant of the inner loop has to be at least 10–100 times smaller than the time constant of the outer loop, otherwise there will be issues with stability. (Zorander 2006) Thought of in another way, the outer loop provides the input to the inner loop, so the inner loop needs to be much more responsive than the outer loop.

It is also usually desirable to control the current of a motor for three reasons. The first is that strain on the motor is reduced by making sure the current curve behaves correctly. Secondly, current control effectively imposes limits on the motor current, preventing the outer controller from trying to drive the motor too hard. And finally, as motor torque is directly related to armature current, the current control loop can be used to implement torque control. (Zorander 2006)

Cascading the three controllers gives a typical control block diagram used for servo applications (see Figure 2.25). This scheme requires two transducers at a minimum—a current sensor and a position sensor.

There are some final issues to consider with closed-loop feedback control of servomotors. It can happen that the motor will start buzzing or twitching about the setpoint. This is because the controller will try to act on fast, small variations in the measured quantities. Therefore high frequency noise filtering of the signals should be done, and a small deadband should be added in the calculation of errors.
Figure 2.25: Block diagram of servo motor with cascaded position, speed and current controllers
Chapter 3

Hardware

3.1 Design Process

This section details the process of designing the BLDC motor controller hardware and schematics to accommodate the new hardware. The design presented in this project essentially follows the considerations outlined in Section 2.4.

Simplified, the hardware design process undertaken was:

- Selection of major components. Before being able to design the new motor controller, it was necessary to decide on the motor and feedback technology. As mentioned previously throughout Chapter 2, BLDC motors and quadrature encoders were selected for use in this project for their various advantages.

- Following selection of the motors, I/O and peripheral requirements were determined and a suitable microcontroller was selected. Detailed consideration to specific parts is discussed further on in Section 3.2.

- Once it was known what the major components of the design were, the schematic design could be produced. The final design presented in this project was mostly derived and pieced together from example schematics found in the various component datasheets, with suitable component sizes calculated for this project. For example, bootstrap components sizes were verified using the equations outlined in Section 2.4.2.

- The schematic remained mostly unchanged from this point throughout the design. Only small changes were made as the circuit was tested to fix minor issues that were encountered, such as pull-up of logic inputs.
3.2 Selection of Major Components

Motors

Because the motors will not be geared down to increase torque, the re-
placement motors should have a relatively high torque rating. When the
Moveaster RM-101 was in production in the early 1980s, high-power rare-
earth magnet BLDC motors were not commercially viable in such a small-
scale, inexpensive robot. According to the Movemaster manual, the largest
of the original stepper motors were rated at just 3.6W. Finding inexpen-
sive BLDC motors that meet or exceed this power rating today is not that
difficult. The largest constraint encountered for this project was not power
or torque requirements, but instead physical size and support for encoder
mounting.

Finding motors that included an extended rear shaft for encoder mount-
ing, that were also not physically too large for the unit, proved difficult.
The Nanotec DBS403 series of BLDC motors were selected primarily be-
cause they were the only suitably sized BLDC motors found that supported
encoder mounting. Fortunately, compatible quadrature encoders are also
provided by Nanotec. The encoders selected were the WEDS5541-B14 se-
ries, which are 3-channel quadrature type encoders with a 1000 pulse per
revolution rating.

Microcontroller

Selection of the microcontroller to be used for the project was essentially
limited to just two aspects:

- On-chip peripherals; and
- Availability of support material

By far the two most well-documented and supported microcontrollers are
the Microchip PIC controllers and the Atmel AVR range. The Microchip
PIC18 series and Atmel ATmega are very popular 8-bit controllers amongst
both hobbyists and enthusiasts, with many textbooks, code examples and
tutorials available for both. Another popular microcontroller is the Texas
Instruments MSP430, though there is considerably more literature avail-
able for the Microchip and Atmel offerings. For this project, the Microchip
PIC18F2331 was specifically selected for the following reasons:

- Six independent on-chip PWM generators and a quadrature decoding
  peripheral. This simplifies the motor controller program considerably
  and frees up system resources, as the quadrature decoding and PWM
  generation can be handled by hardware. Otherwise the PWM gener-
  ation and quadrature decoding would have to be handled in software
by the CPU, which would consume a large amount of instructions, memory and processing time;

- 8MHz internal oscillator;
- I²C communications peripheral allows multiple microcontrollers to be controlled over a common 2-wire bus;
- High-speed 10-bit analogue-to-digital converter (ADC), which is required for analogue current sensing;
- Comprehensive datasheet, and many code examples and applications notes available online regarding BLDC control with PIC18F controllers;
- ICSP allows simple in-circuit programming and debugging of the device;
Inexpensive programming hardware; and  
Free IDE and C compilers provided by Microchip

The Atmel ATmega series was also considered, though no models in this series included on-chip quadrature decoding. To get hardware quadrature decoding using Atmel components would require using the more complex and less documented 32-bit AVR32 architecture. Quadrature decoding on the PIC18F2331 is handled by the Quadrature Encoder Interface (QEI) peripheral.

Speed of the microcontroller was not a large issue with this project as motor control is a relatively slow process. The rated speed of the Nanotec DB42S03 motor is 4000 rpm, which is equivalent to approximately 67 Hz. The encoders, when 4x decoded, have a resolution of 4000 counts per revolution. Therefore when the motor is rotating at its rated speed, the frequency of the encoder counts will be approximately 268 kHz. To effectively control the motor, the microcontroller should operate considerably faster than the speed of the feedback loop. It is not uncommon to find microcontrollers with 8 MHz internal oscillators (the PIC18F series included). One instruction takes four clock cycles to complete, therefore the PIC18F2331 can perform 2 million instructions per second, when operating from a 8 MHz clock source.

**Power Supply**

The transformer in the original unit is rated to 60W at 240VAC. However, as mentioned in Section 3.2, the original stepper motors are only rated at 3.6W or less. This means the peak power demand from the six motors combined would not be expected to be much greater than 20W, leaving around 40W of headroom.  

The Nanotec DB42S03 motors selected for the project are individually rated at the 26W. This is however the rated mechanical power output. The actual rated electrical input power can be calculated by multiplying the rated voltage and current values. For the Nanotec DB42S03, the expected electrical input power in volt-amps \(^1\), under rated conditions, would be 42.96VA.

It was readily apparent that the original transformer would not have enough capacity to support the new motors, and so a new power supply would be needed. It is highly unlikely that all six of the BLDC motors will be running at rated speed and torque at the same time. A diversity factor could be applied to account for the fact that not all motors would be running at such high power at the same time, though this runs the risk that if this scenario were to ever occur, the power supply would likely either trip due

---

\(^1\)Here volt-amps are used instead of watts to differentiate between the motor mechanical power and the electrical input power.
to internal overcurrent protection, or would simply be damaged. Therefore the power supply was sized large enough so that it could supply the peak demand of the entire system.

The required power supply size was determined by multiplying the VA rating of each motor by the number of motors (six), and adding this value to the estimated power draw of the control/logic circuitry. There was no way to really know how much power the control circuit would draw without building the circuit, however a power supply was required to build and test the circuit. Therefore a conservative guess was made regarding the power requirements of the control circuit. The control circuit is assumed to draw 1A at 24V. An efficiency of 80% is also assumed.

<table>
<thead>
<tr>
<th>Motors</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Rated voltage V</td>
<td>24</td>
</tr>
<tr>
<td>Rated current A</td>
<td>1.79</td>
</tr>
<tr>
<td>No. of motors</td>
<td>6</td>
</tr>
<tr>
<td>Power demand</td>
<td>VA 257.76</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Control circuit estimations</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Current A</td>
<td>1</td>
</tr>
<tr>
<td>Efficiency</td>
<td>0.8</td>
</tr>
<tr>
<td>Power demand @24Vdc VA</td>
<td>30</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Total demand</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Total power demand VA</td>
<td>287.76</td>
</tr>
<tr>
<td>Current load @24Vdc A</td>
<td>11.99</td>
</tr>
</tbody>
</table>

Table 3.1: Calculation of required power supply unit

Table 3.1 provides a summary of the assumptions made above. It was determined that the new design would require a 24Vdc power supply rated greater than 290W and with an output current rating greater than 12A. For this project, the TRACOPOWER TXL 300-24S power supply was selected. This unit is rated at 300W and 12.5A.

### 3.3 Circuit Implementation

The BLDC controller circuit consists of a few major components: the microcontroller and logic circuitry, the inverter and gate driver combination, current sensing, and a power stage to supply the different DC voltage levels required throughout the circuit. Figure 3.2 shows a block diagram of how all the elements of complete circuit interface with one another. Complete schematic diagrams are shown in Appendix A.
CHAPTER 3. HARDWARE

Figure 3.2: Block diagram of BLDC controller

The Power Stage

The circuit requires three different DC voltage levels to function—the DC bus for the motor supply is at 24V, the gate driver and MOSFET inverter operate at 15V, and the microcontroller and logic circuitry operate at 5V. Because of this, DC-DC converters are required. Switching step-down regulators are used rather than traditional linear regulators to reduce heat and improve efficiency. A block diagram of the power stage is shown in Figure 3.3.

Figure 3.3: Block diagram of circuit power stage

The 15V regulator selected is a MAX1745. This is an adjustable switch mode regulator that pulse width modulates an external power MOSFET. Voltage feedback from the OUT pin regulates the PWM output to maintain the desired output voltage level. The MAX667 is another switching regulator, however the output is fixed at 5V. Because of the fixed output, the external circuitry is much simpler than the 15V regulator. The only required external component is an output filter capacitor.

The Inverter

An International Rectifier IR2163 gate driver is used for this project. This prefabricated IC integrates three high-side MOSFET drivers and three low-
side drivers, allowing three-phase motor control from a single IC. The advantage of using a gate driver with integrated high and low side drivers is that the inverter can be implemented with all N-channel MOSFETs and a single positive voltage supply, with only a few external bootstrap components required (see Section 2.4.2). The IR2136 also includes a number of motor protection features, which are discussed further on.

The Current Sensor

As mentioned in Section 2.4.3, overload protection is preferable for motor control circuits. The IR2136 gate driver also includes some advanced internal protection, and can shut down the outputs to the inverter MOSFET gates (and therefore disable the motor) if an overcurrent or undervoltage condition is met. The overcurrent condition is triggered when the voltage on the ITRIP pin exceeds 0.46V.

A Hall Effect current sensor was chosen due to the disadvantage of low-side shunt resistive measurement and the relative expense of high voltage op-amps for high-side shunt measurement, as discussed in Section 2.4.3. The current sensor is placed on the high-side to sense the DC link current, and therefore should also be capable of detecting any abnormal ground fault conditions.

The Allegro ACS712 was selected for this project, which has an analogue output and a sensitivity of 185 mV/A. The output is converted to a digital value by the microcontroller ADC and compared to a trip setting in software. The microcontroller sends a logic high (+5V) signal to the ITRIP pin of the gate driver to disable the inverter outputs if an overcurrent condition is detected.

3.4 Controller Prototype

A prototype of the motor controller circuit was implemented on a breadboard, as shown in Figure 3.4. This test bed was also used for testing and verification of the embedded program, as discussed further in Section 4.3. The breadboard prototype differs to the schematics presented in Appendix A in a few ways. The ACS712 current sensor is not implemented, and the 15V is supplied by a linear LM317 linear regulator instead of the proposed MAX1745 switching regulator. The ACS712 and MAX1745 are only available in surface mount packages, therefore making it difficult to test them on a breadboard. Figure 3.4 however does show the disadvantages of using linear regulators for high current applications—large heatsinks must be used to dissipate the generated heat.

After building the prototype it was attempted to test the controller circuit by continually rotating the motor. A simple continual rotation routine was loaded into the microcontroller, however the test was unsuccessful and
the motor did not rotate. As discussed in Section 4.3, the microcontroller outputs and peripherals were tested and proven to be working correctly. This indicates that the problem exists in either the gate driver or inverter circuitry. The inverter circuit is a fairly standard design, and the MOSFETs
were individually tested and proven to switch on and off with a mechanical switch supplying the gate terminals. Testing with a digital multimeter showed that the gate driver outputs were not activating for any input signals. The FAULT pin remained at the pull-up voltage, indicating that undervoltage or overcurrent was not an issue. Replacement of the gate driver IC did not resolve the issue, which suggests the problem exists with either the gate driver external circuitry or the driver IC itself. Further investigation into the IR2136 revealed other users having similar issues without any resolution, however due to time constraints on the project, it was not possible to replace the gate driver IC and redesign the circuit. This issue was left unresolved, however suggestions for possible replacement parts are given in Section 6.2.

3.5 Mechanical Modifications

A large issue encountered in the project was physically adapting the replacement motors to the robot. It was realised early on in the project that it would be near impossible to find suitable replacement motors that had the exact same dimensions and mounting spacing as the original stepper motors used on the robot. This problem is also exacerbated by the fact that the original motors varied in physical dimension and mounting requirements. For the new motors to fit the robot, new mounting plates had to be designed and fabricated. CAD drawings for these mounting plates were drawn up and handed over to be fabricated.

The drive gears also had to be removed from the old stepper motors and fit onto the replacement motors. Further issues were encountered when it was realised that some of the old motors had larger rotor diameters than the new motors. Because of this, some of the drive gears did not fit the new motors and sleeves also had to be fabricated. Figure 3.5 shows the robot reconstructed with the new Nanotec BLDC motors and encoders mounted.

An issue realised with the new design is the lack of position stability when there is no power supplied to the motors. The original motors had gearboxes attached to increase torque applied to the output shafts. The friction of the gearboxes also acted to prevent the joints from moving under their own weight once they are actuated to a particular position. Because the new motors are directly driven without gearboxes, the weight of each link coupled with the spring tension on some joints produces a torque that automatically collapses the robot back to a particular position.

This means that the robot will be unable to maintain a user specified position without power to the motors to overcome this torque. This could be a problem if the device is being used for a critical application. The rapid change in position when power is lost could damage either the robot or the payload. For this reason, when the robot is operational, it should be manually returned to this position by the user before the power supply is
When reconstructing the robot, it was also found that the lack of gearbox friction presented a problem with tensioning the finger-grip pulley system. The pulley cable could not be wound around the winch like it was in the original unit, as it would unwind and lose tension immediately. The cable had to be wound differently to how it was previously, and now relies on one of the rear springs to maintain tension on the cable. However this spring is not strong enough to completely open the finger grips, and should be replaced by a stronger spring to improve the operating range of the finger grips. Another (better) improvement would be to implement an actual cable tensioning system for this cable.
Chapter 4

Embedded Program

The main program was written in C, using the MPLAB IDE and MPLAB C18 compiler. These were selected as they are provided free of charge by Microchip, and have good support for the PICkit 2 hardware programmer that was used. Specifically MPLAB IDE v8.84 and MPLAB C18 v3.40 were used.

The main routine is shown in Figure 4.1. The initialisation routine is immediately called, which configures the hardware peripherals. These include the quadrature encoder interface, the power control PWM module, the ADC, and the Synchronous Serial Port (used for I2C communications). Following initialisation of the peripherals the IR2136 gate driver is enabled. The program then enters an infinite loop, where a control routine is continually called. This routine performs the main commutation and feedback position control of the system.

4.1 Control Routine

This section outlines a proposed control strategy to implement a closed-loop position control routine using the hardware and peripherals outlined earlier. The routine was programmed and presented as part of the final firmware outlined in Appendix B, however is still largely untested due to not having a working hardware prototype. The control routine uses the QEI and PWM modules of the PIC18F to control the output speed depending on the position error. As the position of the joint approaches the setpoint, the velocity will decrease and stop when it is within a deadband region. The control routine is shown in Figure 4.2.

The number of encoder counts is directly related to the mechanical displacement of the rotor shaft $\theta_m$ and the encoder resolution (counts per degree of rotation). The actual angular output of the joint $\theta_j$ is related to the rotor displacement by the gear reduction ratio $R$ for that joint. Even though a gearbox is not used, a gear reduction ratio still arises because of
the planetary gearing arrangement of the joint (the ratio between the motor drive gear and the fixed cog that the joint rotates about).

\[ \theta_j = \theta_m \times R \] (4.1)

Therefore the number of encoder counts \( n \) expected for a specified change in joint displacement is given by the equation:

\[ n = \theta_m \times R \times \text{resolution} \] (4.2)

When a movement setpoint is received in degrees, the program converts this value to encoder counts as per Equation 4.2. This value is stored as an integer setpoint. The current number of encoder counts is stored in the POSCNT register. The position error is calculated as the difference between the setpoint integer and the POSCNT register, which is then compared to a deadband limit. If the error is within the deadband, the PWM outputs are disabled and the control routine exits, otherwise the error is passed onto the PI controller. The PI controller calculates the speed reference, which is used to determine the duty cycle of the PWM outputs.

The duty cycle for a pair of PWM outputs is determined by the value loaded into the PDCx registers. The maximum (100%) duty cycle value is
equal to four times the value stored in the PTPER register. Therefore the duty cycle is calculated by Equation 4.3 below.

\[
\text{Duty cycle} = \frac{\text{PTPER} \times 4}{\text{Max. speed ref}} \times \text{speed reference} \quad (4.3)
\]

Once the duty cycle is loaded, the appropriate pair of PWM outputs need to be enabled. The status of the Hall sensors are read and the correct commutation table is loaded based upon the desired direction of rotation. The commutation table is predefined and enables two PWM outputs based upon the Hall sensor status. Individual PWM outputs are enabled by toggling the relevant bit in the OVDCOND register. For example, setting the fifth bit of the OVDCOND register to 1 enables the PWM5 output of the PIC.

At this point the correct phase pair should be enabled with the duty cycle determined by the closed-loop PI routine. The program loops this routine until the error is within the position deadband limit.

### 4.2 Interrupt Service Routine

When a high priority event occurs, an interrupt service routine (ISR) is called. Interrupts are high priority flags generated by external peripherals to notify the main CPU that something has happened that the CPU should respond to. For the PIC18F the interrupt handler is located at the CPU memory address 008h. When an interrupt is generated, the CPU saves its current state, runs the program stored at this address, and then returns to its previous position in the main program.

Two hardware interrupts have been configured—one from the ADC and another from the I²C peripheral. The ADC interrupt is generated every time a new conversion is completed. The ISR compares the result to the overcurrent trip setting. If the conversion result is greater than the trip setting, the PWM outputs are disabled and the ITRIP input of the IR2136 is forced high, therefore inhibiting any voltage to be applied to the motor phases. The PWM outputs and gate driver are re-enabled when the ADC records a current reading below the trip point.

The I²C interrupt is generated upon each byte received. The ISR checks if the last byte received was a hardware address or actual data. This is done by checking the D/Â bit of the SSPSTAT register. If the last byte was data, it is assumed to be a new position setpoint. The ISR converts the value into the nearest encoder count integer and clears the previous position counter value. The PI controller is reset, interrupt flags are cleared and the ISR exits.
CHAPTER 4. EMBEDDED PROGRAM

Figure 4.2: Proposed control strategy
CHAPTER 4. EMBEDDED PROGRAM

Figure 4.3: Interrupt service routine
4.3 Proof of Operation

While a working prototype of the final design was not completed, certain subsystems were tested and proved to be working. A test circuit was constructed on a breadboard to test the initialisation of the peripherals (see Figure 3.4). The following testing was undertaken to prove the operation of the peripherals.

- The quadrature encoder was connected to the microcontroller and the rotor shaft was turned by hand. Monitoring the \texttt{POS_CNT<:L>} registers in MPLAB showed that the encoder counter incremented and decremented as the rotor was turned.

- PWM outputs were tested with LEDs. The PWM outputs are configured as independent outputs so that each output can be individually enabled or disabled. The PWM outputs are connected in groups of two (PWM0/1, PWM2/3 and PWM4/5). During testing with LEDs, it was realised both outputs of the same group must have the same duty cycle, therefore the schematic was rearranged and the program firmware adjusted so that the high and low side outputs of each phase belong to the same PWM group. Because the high and low side FETs of each phase are never turned on at the same time, it is safe to apply the same duty cycle to both outputs. LED brightness could be successfully adjusted with the PWM module, indicating the PWM outputs are configured correctly.

- Hall sensor feedback was verified. Depending on the Hall sensor state, the pair of active LEDs connected to the PWM outputs changed.

- A potentiometer was connected to pin AN1 as a variable voltage divider, so that the voltage present at the ADC input could be adjusted. Monitoring the \texttt{adc_result} integer showed that the result changed as the potentiometer was varied, therefore verifying the ADC configuration and interrupt service routine.

These results prove that the ADC, QEI and PWM peripherals are correctly configured and working, and that the commutation routine functions correctly. The I²C communications and PI controller aspects of the firmware however can not be tested and verified until a working hardware prototype is constructed.
Chapter 5

Simulink Model

A preliminary model of the system was carried out using MATLAB Simulink. The original intention was to compare the response of the final robotic manipulator system to a theoretical model. However, to do this a complete electromechanical model of the system would be required, and so this aspect of the project was abandoned. The results of the initial modelling exercise are included here as a reference for anyone who wishes to build upon the work carried out in this project. Here, only the BLDC motor and the three-phase inverter are modelled. Extensive literature is readily available on modelling wye-connected BLDC motors, however to the author’s knowledge, not much work has been done with the analysis and modelling of delta-connected motors.

The combined inverter/motor model is shown below in Figure 5.1. This model essentially simulates the BLDC motor in open-loop configuration. The supply voltage $V_s$ and applied torque can be changed to see the effects on phase currents, motor speed and torque.

Figure 5.1: Simulink model of BLDC motor and 3-phase inverter in open-loop configuration

5.1 BLDC Motor Model

To model the delta-connected BLDC motor a state-space phase variable approach is chosen. Neglecting mutual inductance and assuming that all the
stator resistances and inductances are equal, the motor can be described by the following equations:

\[
V_{ab} = R_{i_a} + L \frac{di_a}{dt} + e_a 
\]
(5.1)

\[
V_{bc} = R_{i_b} + L \frac{di_b}{dt} + e_b 
\]
(5.2)

\[
V_{ca} = R_{i_c} + L \frac{di_c}{dt} + e_c 
\]
(5.3)

\[
T_e = J \frac{d\omega_m}{dt} + \beta \omega_m + T_L 
\]
(5.4)

\[
\omega_m = \frac{d\theta_m}{dt} 
\]
(5.5)

The symbols \(V\), \(i\) and \(e\) denote the phase-to-phase voltages, currents and back EMFs respectively, and the subscripts \(a\), \(b\) and \(c\) are used to indicate the three phases. \(R\) and \(L\) denote the stator per-phase resistance and inductance, while \(T_e\) and \(T_L\) are the electrical torque and load torque. Finally \(J\) is the rotor inertia, \(\beta\) is the friction coefficient, \(\omega_m\) is the rotor speed and \(\theta_m\) is the rotor mechanical position.

The back EMFs of the motor are functions of the rotor velocity and the back EMF constant. As the motor is delta-connected, it is assumed the back EMF will be sinusoidal rather than trapezoidal. It is also assumed the three back EMFs are balanced and 120° phase-shifted from one another. The three back EMFs are then described by the equations:
Here the term $\theta_e$ denotes the rotor electrical position, which is related to the mechanical position by the function $\theta_e = \frac{p}{2} \theta_m$, where $p$ is the number of poles. $K_e$ is the back EMF constant of the motor. Typically PM motors with sinusoidal back EMF are modelled using a rotating d-q axis model. This method however does not lend itself well to motors with trapezoidal back EMF, hence the use of a phase variable model. (Krishnan 2001) Theoretically this model would also be valid for delta-connected BLDC motors with trapezoidal back EMF by replacing the sine component of the above functions with trapezoidal functions.

Finally, the electrical torque is related to the phase currents through the torque constant $K_t$ and the equation:

$$T_e = K_t \left( i_a \sin(\theta_e) + i_b \sin \left( \theta_e - \frac{2\pi}{3} \right) + i_c \sin \left( \theta_e + \frac{2\pi}{3} \right) \right)$$  (5.9)

The back EMF and electrical torque equations cannot be separated such that the phase variables are linearly independent. Therefore these terms are derived outside of the state-space system and fed back into the model as part of the control vector. Rearranging equations 5.1–5.5 into a state-space representation gives the following model of the motor:

$$\begin{bmatrix}
    \dot{i}_a \\
    \dot{i}_b \\
    \dot{i}_c \\
    \dot{\omega}_m \\
    \dot{\theta}_m
\end{bmatrix} = \begin{bmatrix}
    \frac{-R}{L} & 0 & 0 & 0 & 0 \\
    0 & \frac{-R}{L} & 0 & 0 & 0 \\
    0 & 0 & \frac{-R}{L} & 0 & 0 \\
    0 & 0 & 0 & -\frac{\beta}{J} & 0 \\
    0 & 0 & 0 & 1 & 0
\end{bmatrix} \begin{bmatrix}
    i_a \\
    i_b \\
    i_c \\
    \omega_m \\
    \theta_m
\end{bmatrix} + \begin{bmatrix}
    1 \\
    0 \\
    0 \\
    0 \\
    0
\end{bmatrix} \begin{bmatrix}
    \frac{1}{L} & 0 & 0 & 0 \\
    \frac{1}{L} & 0 & 0 & 0 \\
    0 & \frac{1}{L} & 0 & 0 \\
    0 & 0 & \frac{1}{J} & 0 \\
    0 & 0 & 0 & 0
\end{bmatrix} \begin{bmatrix}
    V_{ab} - e_a \\
    V_{bc} - e_b \\
    V_{ca} - e_c \\
    T_e - T_L
\end{bmatrix}$$  (5.10)

$$\begin{bmatrix}
    i_a \\
    i_b \\
    i_c \\
    \omega_m \\
    \theta_m
\end{bmatrix} = \begin{bmatrix}
    1 & 0 & 0 & 0 & 0 \\
    0 & 1 & 0 & 0 & 0 \\
    0 & 0 & 1 & 0 & 0 \\
    0 & 0 & 0 & 1 & 0 \\
    0 & 0 & 0 & 0 & 1
\end{bmatrix} \begin{bmatrix}
    i_a \\
    i_b \\
    i_c \\
    \omega_m \\
    \theta_m
\end{bmatrix}$$  (5.11)
The contents of the BLDC block are shown in Figure 5.3. The state-space system implements equations 5.10 and 5.11 using Simulink’s inbuilt state-space block. The outputs of the state-space system are the phase currents, rotor speed and mechanical position. The contents of ‘Subsystem’ are shown in Figure 5.4. This block transforms the rotor mechanical position into the electrical position. The lookup table produces an integer between 0 and 5 depending on the rotor electrical position, which is sent to the inverter via the Goto tag. The purpose of the lookup table is to simulate the Hall effect sensors, which have six possible outputs and are used to switch the commutation state in the inverter block. From the rotor electrical position, the subsystem also creates a vector called $\text{sin}_{\text{abc}}$, which contains three sine waves that are phase shifted by $120^\circ$ from one another. The $\text{sin}_{\text{abc}}$ vector is multiplied by the back EMF constant and the rotor speed to obtain the three back EMFs. The back EMFs are then also sent to the inverter via a Goto tag, where they are used to determine the phase-to-phase voltage inputs for the state-space system. This is discussed in further detail in Section 5.2.

The electrical torque is calculated as per equation 5.9. This equation is the dot product of the current vector $I_{\text{abc}}$ and the $\text{sin}_{\text{abc}}$ vector, multiplied by the torque constant. The electrical torque is then fed back to the input of the state-space system, where the load torque is subtracted to produce the input term $T_e - T_L$.

![Figure 5.3: Contents of BLDC motor block](image)

### 5.2 Three-Phase Inverter

A model of the three-phase inverter is required to simulate the discontinuous voltages that are applied to the phase windings of the motor. To model the inverter, the MOSFETs are assumed to be ideal; they are treated as ideal switches and switching and conduction losses are ignored. The inverter block then simply determines the phase-to-phase voltage relationship depending on the commutation state and applies these voltages to the BLDC motor.
Figure 5.4: Calculation of position and sine wave vector in Subsystem block. The method used to model the inverter here has been adapted from the work done by Baldursson (2005) to suit the delta configuration.

Consider the delta-connected configuration during commutation sequence AB, as shown in Figure 5.5. During the sequence AB it is obvious that the terminal voltages of phases A and B will be equal to $V_s$ and 0 respectively. The terminal voltage of phase C can be determined by applying Kirchoff’s voltage law and analysing the voltages in the phase B–C branch.

Figure 5.5: Equivalent circuit for delta-connected BLDC motor during commutation state AB

From inspection of Figure 5.5 the following equations can be determined:

\[ V_s = R_i c + L_i \frac{di_c}{dt} - e_c + R_i b + L_i \frac{di_b}{dt} - e_b \]  \hspace{1cm} (5.12)

\[ i_c = i_b \]  \hspace{1cm} (5.13)

It is also apparent that the terminal voltage of phase C will be equal to
the sum of the voltages in phase B.

\[ V_c = R_i b + L \frac{di_b}{dt} - e_b \]  

(5.14)

By combining equations 5.12, 5.13 and 5.14, the terminal voltage of phase C can be expressed by the equation:

\[ V_c = \frac{1}{2} (V_s - e_b + e_c) \]  

(5.15)

Knowing the three terminal voltages, the following phase-to-phase voltage relationships can be determined:

\[ V_{ab} = V_s \]  

(5.16)

\[ V_{bc} = \frac{1}{2} (-V_s + e_b - e_c) \]  

(5.17)

\[ V_{ca} = \frac{1}{2} (-V_s - e_b + e_c) \]  

(5.18)

The effective voltage seen by each stator winding will be the difference between the phase-to-phase voltage and the back EMF for that phase. This subtraction could be done in the BLDC motor block at the input to the state-space system. However as the inverter must know the back EMFs to calculate the phase-to-phase voltages, it is more computationally effective to simply subtract the back EMFs from the inverter output equations. For the commutation sequence AB then, the output equations will be:

\[ V_{ab} - e_a = V_s - e_a \]  

(5.19)

\[ V_{bc} - e_b = \frac{1}{2} (-V_s - e_b - e_c) \]  

(5.20)

\[ V_{ca} - e_c = \frac{1}{2} (-V_s - e_b + e_c) \]  

(5.21)

The resulting voltages for all six commutation states are shown below in Table 5.1. This table is implemented in the model as a Simulink S-function inside the inverter block. The function receives the current commutation sequence and back EMFs from the BLDC motor block, along with the DC input voltage and uses a simple switch case statement to determine the three output voltages.
CHAPTER 5. SIMULINK MODEL

<table>
<thead>
<tr>
<th>Sequence</th>
<th>$V_{ab} - e_a$</th>
<th>$V_{bc} - e_b$</th>
<th>$V_{ca} - e_c$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0: AB</td>
<td>$V_s - e_a$</td>
<td>$\frac{1}{2} (-V_s - e_b - e_c)$</td>
<td>$\frac{1}{2} (-V_s - e_b - e_c)$</td>
</tr>
<tr>
<td>1: AC</td>
<td>$\frac{1}{2} (V_s - e_a - e_b)$</td>
<td>$\frac{1}{2} (V_s - e_a - e_b)$</td>
<td>$V_s - e_c$</td>
</tr>
<tr>
<td>2: BC</td>
<td>$\frac{1}{2} (-V_s - e_a - e_c)$</td>
<td>$V_s - e_b$</td>
<td>$\frac{1}{2} (-V_s - e_a - e_c)$</td>
</tr>
<tr>
<td>3: BA</td>
<td>$-V_s - e_a$</td>
<td>$\frac{1}{2} (V_s - e_b - e_c)$</td>
<td>$\frac{1}{2} (V_s - e_a - e_c)$</td>
</tr>
<tr>
<td>4: CA</td>
<td>$\frac{1}{2} (-V_s - e_a - e_b)$</td>
<td>$V_s - e_b$</td>
<td>$\frac{1}{2} (-V_s - e_a - e_c)$</td>
</tr>
<tr>
<td>5: CB</td>
<td>$\frac{1}{2} (V_s - e_a - e_c)$</td>
<td>$-V_s - e_b$</td>
<td>$\frac{1}{2} (V_s - e_a - e_c)$</td>
</tr>
</tbody>
</table>

Table 5.1: Inverter output voltages

5.3 Results

The model was simulated using the parameters shown in Table 5.2. These parameters were taken from the Nanotec DBS403 datasheet to simulate the motors at rated conditions. The velocity and torque responses of the model are shown in Figures 5.6 and 5.7 respectively.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of poles</td>
<td>8</td>
</tr>
<tr>
<td>Nominal voltage</td>
<td>24</td>
</tr>
<tr>
<td>Resistance per phase</td>
<td>1.5</td>
</tr>
<tr>
<td>Inductance per phase</td>
<td>H</td>
</tr>
<tr>
<td>Voltage constant</td>
<td>V·s/ rad</td>
</tr>
<tr>
<td>Torque constant</td>
<td>N·m/A</td>
</tr>
<tr>
<td>Rotor inertia</td>
<td>kg·m²</td>
</tr>
<tr>
<td>Friction constant</td>
<td>N·m·s</td>
</tr>
<tr>
<td>Nominal load torque</td>
<td>N·m</td>
</tr>
</tbody>
</table>

Table 5.2: Simulink simulation parameters

The two curves tend to follow the first-order response expected from a DC motor, however analysis of the velocity curve shows some issues with model. The model was simulated using the motor constants and rated values from the vendor datasheet, however at rated voltage and torque, the output speed was 530 rad/s. This is equivalent to 5061 rpm, which is 1000 rpm greater than the expected output as per the rated speed mentioned in the datasheet (4000 rpm). Nonetheless, what the model does highlight is the infamous torque ripple problem, which is clearly evident in the output graphs. The drastic variations in output torque result in an unsteady final speed. There also appears to be a dramatic dip in output velocity and torque at around 0.005 seconds.
Figure 5.6: Velocity response of delta-connected BLDC motor model

Figure 5.7: Torque response of delta-connected BLDC motor model
Chapter 6

Discussion

6.1 Summary

The project did not come to an end where all the original objectives outlined during the planning phase were completed. It was initially intended that the end deliverables of the project would be a completely functional robotic arm and a PC interface to control it. This was reduced to just the upgrade of robotic arm itself due to time constraints on the project, however further issues were encountered with successfully implementing a prototype BLDC motor drive circuit. The lack of a working hardware prototype made any further advancement on the project difficult. Without a working motor drive circuit, it was impossible to verify all aspects of the embedded microcontroller firmware and to implement the closed-loop position control routines.

Regardless, the major practical achievements of the project were:

- The robot was reconstructed with new BLDC motors with position encoders attached. The unit is mechanically functional, though a working electrical control circuit was not completed. The new hardware however includes the capacity for closed-loop position, speed and current control, which was the main objective of the project.

- Detailed electrical schematics were produced and microcontroller firmware programming was completed. Issues were identified with the gate driver and bootstrap circuitry, however the schematics presented in this project provide a starting point for any future improvements. In the Further Work section, suggestions are made as to how the issues involving the gate drivers could be resolved. The firmware presented in this project has also been tested and proven to work as far as possible without working motor drive circuitry.

- Capacity for a completely new, more user-intuitive control interface to be implemented has been added to the robot by the selection of ad-
advanced PIC18F microcontrollers. The new microcontrollers support a variety of communication protocols such as I\(^2\)C, SPI or RS232, opening up the possibility for a completely custom HMI to be designed. Preliminary firmware programming was carried out to initialise the I\(^2\)C communications peripheral on the microcontrollers, simplifying the future implementation of an advanced PC-based HMI.

Despite the limited outcomes, the remaining work for completion of the project has been planned in detail, possibly as another final year engineering project. Section 6.2 provides a detailed breakdown of the work remaining and suggestions for how stage of the remaining work could be implemented.

### 6.2 Further Work

There is still a large amount of work remaining before the overhaul of the robot is complete. The major items remaining are outlined in this section. Most of the items here were initially considered for this project, with solutions suggested and planned, however were not implemented due to the time constraints of the project and deliverables.

The first step in completing the upgrade will be to build a working prototype of the BLDC controller outlined in the above sections. As mentioned earlier, the circuit was constructed on a breadboard but issues were encountered getting the MOSFETs to turn on. Further investigation and troubleshooting needs to be done into this issue. An alternative solution would be to replace the IR2136 gate driver with separate gate drivers for each motor phase. The IR2101 gate driver has been successfully used in other projects such as the OpenBLDC project, and may be a suitable replacement for this project. This chip combines a single high side and low side driver and removes the motor protection features (undervoltage and overcurrent shutdown). The disadvantage of this method is that the chip count and PCB footprint are significantly increased as three IR2101s must be used for each motor.

The next step would be to verify the position control routine suggested in Section 4.1. To smooth out the response of the system, the control routine could be improved to use cascaded controller. The Allegro ACS712 provides link current feedback, allowing closed loop current control. The QEI peripheral on the PIC18F also allows for velocity measurement from the encoder. Theoretically all the hardware required for a triple-cascade system as per Section 2.5.2 is provided in this design. All that is required is to implement the routine in the microcontroller firmware. Alternatively, the simple PI speed controller and deadband combination suggested in this project could be kept. Improving on the Simulink model presented in Chapter 5 could assist with selecting the optimal gain values, or a trial-and-error approach could also be taken. The ZN method is another possibility for PI tuning.
A factor that might need to be considered when implementing a feedback control system, especially a cascaded one, is the oscillator frequency of the microcontroller. The maximum internal oscillator frequency of 8 MHz may not be fast enough for effective control of the motor at high speeds, however the PIC18F2331 allows external crystal oscillators up to 20 MHz to be used as the main clock source. The pins required for this (pins 9 and 10) have been left unused in the design presented in this project in case an external oscillator is required for future improvements. The PIC18F2331 datasheet from Microchip provides extensive information on connecting and configuring these oscillators.

After a working prototype has been constructed and tested, the next step would be to complete the communications and HMI design. This is potentially the largest part of the remaining work. The design outlined in this project assumes the I²C protocol will be used for the communicating with the HMI. The pins associated with the I²C communication peripheral on the PIC18F2331 (pins 15 and 16) have been configured for I²C communications and the required pull-up resistors added to the schematic. To complete the communications interface an I²C to USB converter is required. The FTDI FT2232H is one such chip. FTDI also provide free software drivers for Microsoft Windows, allowing an application running on a local PC to communicate with the PIC microcontrollers over USB via the FT2232H. The document ‘Interfacing FT2232H Hi-Speed Devices To I²C Bus’ from FTDI provides an extensive example of interfacing with I²C devices over USB using the FT2232H. For the HMI, the National Instruments LabVIEW software was planned to be used. The software includes the ability to communicate with devices connected to a USB port on the computer, therefore a custom LabVIEW interface could be designed to control and monitor each motor on the robot.

Finally a printed circuit board needs to be designed and built. The circuit board will need to include the power stage, six BLDC controllers and a communications interface.
References

Åström, KJ 2002, Control systems design lecture notes for ME 155A, Department of Mechanical and Environmental Engineering, University of California, Santa Barbara, Unpublished.
Akin, B & Bhardwaj, M 2010, Trapezoidal Control of BLDC Motors Using Hall Effect Sensors, Texas Instruments application notes.
Elliott, C & Bowling, S 2004, Using the dsPIC30F for Sensorless BLDC Control, Microchip application notes, AN901.
REFERENCES

Lepkowski, J 2003, Motor Control Sensor Feedback Circuits, Microchip application notes, AN894.
Willis, MJ 1999, Proportional-Integral-Derivative Control, Dept. of Chemical and Process Engineering, University of Newcastle, Unpublished.
Bibliography

Åström, KJ 2002, *Control systems design lecture notes for ME 155A*, Department of Mechanical and Environmental Engineering, University of California, Santa Barbara, Unpublished.


Jani, Y 2006, *Implementing Embedded Speed Control for Brushless DC Motors, Part 1*, Renesas Technology America, Inc.


Appendix A

Schematic Diagrams
Figure A.1: Power stage schematic diagram
Figure A.2: BLDC controller schematic diagram
Appendix B

Firmware

#include <p18f2331.h>
#include <delays.h>
#include <math.h>

#pragma config WDTEN = OFF // Disable watchdog timer
#pragma config OSC = IRCIO // Internal oscillator, I/O on RA6 & RA7
#pragma config LVP = OFF // Disable low voltage programming
#pragma config HPOL = LOW // PCPWM active-low polarity
#pragma config LPOL = LOW

// Configurable controller parameters
#define KP 0.1
#define KI 0.1
#define IMAX 100
#define IMIN -100
#define DEADBAND 10
#define TRIP 0xCC
#define GEAR_RATIO 1

// Declarations
void high_isr(void);
volatile float sp;
volatile float Err;
volatile float iErr;
volatile float PIout;

// Initialisation routine
void init(void)
{
    OSCCON |= 0x70; // 8MHz oscillator frequency
APPENDIX B. FIRMWARE

RCONbits.IPEN = 1;  // Enable interrupt priorities
INTCONbits.GIEH = 1;  // Global interrupts enabled
INTCONbits.GIEL = 1;  // Peripheral interrupts enabled

//------------------------------------------------------------------------------
// I/O ports
//------------------------------------------------------------------------------
TRISA = 1;  // PORT A I/O direction (11111111)
TRISB = 0;  // PORT B I/O direction (00000000)
TRISC = 0xB7;  // PORT C I/O direction (10110111)

PORTA = 0;  // All I/O ports initialised in off state
PORTB = 0;
PORTC = 0;

//------------------------------------------------------------------------------
// Motion feedback module
//------------------------------------------------------------------------------
DFLTCON |= 0x78;  // Enable noise filters
POSCNTL = 0;  // Clear position counter
POSCNTH = 0;

MAXCNTL = 0xFF;  // Set maximum position count to 0xFFFF (65535)
MAXCNTH = 0xFF;

QEICONbits.QEIM = 0x06;  // QEI enabled in 4x decoding mode

//------------------------------------------------------------------------------
// PCPWM generators
//------------------------------------------------------------------------------
PTCON0 = 0;  // PWM timer in free-running mode

PTPERL = 0x6E;  // PWM frequency 18kHz (8MHz osc, 1:1 prescale)
PTPERH = 0;

PWMCON0 = 0x47;  // PWM0-5 outputs enabled in independent mode
PWMCON1 = 0x01;  // Output overrides synced wrt PWM timebase
DTCON = 0;  // No deadtime inserted

OVDCOND = 0;  // All PWMs off upon initialisation
OVDCONS = 0;

FLTCONFIG = 0;  // Fault inputs disabled
SEVTCMPL = 0; // Special event trigger not used
SEVTCMPH = 0;

PDC0L = 0; // Clear all PWM duty cycle registers
PDC1L = 0;
PDC2L = 0;
PDC0H = 0;
PDC1H = 0;
PDC2H = 0;

PTCON1 = 0x80; // PWM timer on

().'/------------------------------------------------------------------
// Synchronous serial port (for I2C)
.'/------------------------------------------------------------------
SSPADD = 0x10; // Unique device address (0x10)
SSPCON = 0x16; // I2C slave mode, 7-bit address
PIE1bits.SSPIE = 1; // SSP interrupt enabled
PIR1bits.SSPIE = 0; // Clear SSP interrupt flag on init
IPR1bits.SSP1IP = 1; // SSP interrupt set to high priority
SSPCONbits.SSPEN = 1; // Enable serial port

.'/------------------------------------------------------------------
// A/D converter
.'/------------------------------------------------------------------
ANSEL0 = 0x1; // Disable all analog inputs except AN0
ADCHS &= 0xFC; // AN0 selected as input channel
ADCON0 = 0x20; // Continuous loop mode, single channel mode,
// group A selected
ADCON1 = 0x10; // AVdd and AVss pins selected as vref,
// FIFO buffer enabled
ADCON2 = 0x94; // A/D result right justified, 4TAD sample time,
// Fosc/4 conversion clock
ADCON3 = 0x80; // Interrupt generated on 4th word write to buffer,
// all triggers disabled
PIE1bits.ADIE = 1; // A/D converter interrupt enabled
PIR1bits.ADIF = 0; // Clear A/D interrupt flag
IPR1bits.ADIP = 1; // A/D converter interrupt set to high priority
ADCON0bits.ADON = 1; // A/D converter on
Delay10TCYx(2); // Power-up delay of 20 cycles (~10us w/ 8MHz osc)
ADCON0bits.GO = 1; // Start A/D converter
}

// PI controller main routine
void PImain(void)
{
  /*
  PI output calculation.
  Read latest encoder count value, calculate errors, check if
  the error is within deadband limit, calculate PI output.
  */

  unsigned int pos = POSCNTH;
pos = (pos << 8) | POSCNTL;
Err = sp - pos;

  if(fabs(Err) > DEADBAND){
    iErr += Err;

    // anti-windup limiter
    if(iErr>IMAX){
      iErr = IMAX;
    }else if (iErr<IMIN){
      iErr = IMIN;
    }

    PIout = KP * Err + KI * iErr;
  }else{
    PIout = 0;
  }

  /*
  Calculate the duty cycle from the PI controller output.
  First calculate floating point duty cycle, then round to the
  nearest integer, and finally load value into the two duty
  cycle registers PDCx<L:H>.

  DUTY_CYCLE = (PTPER*4 / MOTOR_RATED_SPEED) * SPEED_REFERENCE
  Therefore DUTY_CYCLE = 440/4000 * SPEED_REFERENCE + 0.5
  0.5 added to assist with rounding float to int
  */
double duty_f = 0.11 * fabs(PIout) + 0.5f;
int duty = (int) floor(duty_f);

unsigned char dutyH = duty & 0xff00;
unsigned char dutyL = duty & 0x00ff;

PDC0L = dutyL;
PDC0H = dutyH;
PDC1L = dutyL;
PDC1H = dutyH;
PDC2L = dutyL;
PDC2H = dutyH;
}

// Commutation routine
void commutation(void)
{
    // isolate hall sensor status bits
    unsigned char hall = PORTC & 0x07;

    /*
     * Commutation sequence begins here. First check if position error
     * is negative. Negative result means reverse rotation is required.
     * Enable PWM outputs based on hall sensor status.
     */

    if(Err < 0){
        // reverse rotation
        switch (hall){
            case 0x01:
                OVDCOND = 0x18;
                break;
            case 0x03:
                OVDCOND = 0x09;
                break;
            case 0x02:
                OVDCOND = 0x21;
                break;
            case 0x06:
                OVDCOND = 0x24;
                break;
            case 0x04:
                OVDCOND = 0x06;
break;
case 0x05:
    OVDCOND = 0x12;
    break;
default:
    // invalid input, turn off all PWM outputs
    OVDCOND = 0;
    break;
}
}
else{
    // forward rotation
    switch (hall){
    case 0x01:
        OVDCOND = 0x21;
        break;
    case 0x03:
        OVDCOND = 0x24;
        break;
    case 0x02:
        OVDCOND = 0x06;
        break;
    case 0x06:
        OVDCOND = 0x12;
        break;
    case 0x04:
        OVDCOND = 0x18;
        break;
    case 0x05:
        OVDCOND = 0x09;
        break;
    default:
        // invalid input, turn off all PWM outputs
        OVDCOND = 0;
        break;
    }
    }
}

// Main routine
void main(void)
{
    init();
// enable gate driver
PORTCbits.RC3 = 1;

// main loop begins here
while(1){
    P1main();
    commutation();
}

// High priority interrupt vector
#pragma code high_vector=0x08
void interrupt_at_high_vector(void)
{
    _asm
        GOTO high_isr
    _endasm
}

// High-priority ISR
#pragma code
#pragma interrupt high_isr
void high_isr(void)
{
    if (PIR1bits.ADIF){
        // get new adc result
        unsigned int adc_result = ADRESH;
        adc_result = (adc_result << 8) | ADRESL;
        // compare to trip setting
        if (adc_result < TRIP){
            PORTCbits.RC6 = 0;
        } else {
            PORTCbits.RC6 = 1;
        }
        // clear flags
        PIR1bits.ADIF = 0;
    }

    if (PIR1bits.SSPIE){
        if (SSPCONbits.DA){
            // last byte was data, convert setpoint
            sp = SSPBUF * 11.11 * GEAR_RATIO;
            // reset PI controller
        }
    }
}
Err = 0;
iErr = 0;
PIout = 0;
// reset encoder count
POSCNTL = 0;
POSCNTH = 0;
}
// clear flags
PIR1bits.SSPIE = 0;
}
Appendix C

Nanotec DB42S03 datasheet

Courtesy of Nanotec GmbH & Co KG.
Appendix D

Fourier expansion of trapezoidal back EMF

The trapezoidal wave is a discontinuous function that consists of five linear segments over a complete cycle. Determining the expression for each linear segment using the straight line equation \( y = mx + c \) yields the following trapezoidal function:

\[
\begin{align*}
  f(x) &= \begin{cases} 
    -6 - \frac{6x}{\pi} & \text{if } -\pi \leq x < -\frac{5\pi}{6} \\
    -1 & \text{if } -\frac{5\pi}{6} \leq x < -\frac{\pi}{6} \\
    \frac{6x}{\pi} & \text{if } -\frac{\pi}{6} < x \leq \frac{\pi}{6} \\
    1 & \text{if } \frac{\pi}{6} \leq x < \frac{5\pi}{6} \\
    6 - \frac{6x}{\pi} & \text{if } \frac{5\pi}{6} < x \leq \pi
  \end{cases}
\end{align*}
\]

By definition, the Fourier series is:

\[
f(x) = a_0 + \sum_{n=1}^{\infty} (a_n \cos(nx) + b_n \sin(nx))
\]

The coefficients \( a_0, a_n \) and \( b_n \) are determined by the following formulas:
APPENDIX D. FOURIER EXPANSION OF TRAPEZOIDAL BACK EMF

\[ a_0 = \frac{2}{\pi} \int_{-\pi}^{\pi} f(x) \, dx \]
\[ a_n = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \cos(nx) \, dx \]
\[ b_n = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \sin(nx) \, dx \]

To find the coefficients of the series, first consider the symmetry of the trapezoid. The trapezoidal waveform function is an odd function. Odd functions are those that meet the criteria \(-g(x) = g(-x)\). If a function is odd, all the cosine coefficients \((a_n)\) will equal zero, and the sine coefficients \((b_n)\) will be equal to twice the value of integrating half the range. Therefore:

\[ b_n = \frac{2}{\pi} \int_{0}^{\pi} f(x) \sin(nx) \, dx \]
\[ = \frac{2}{\pi} \int_{0}^{\pi/6} \left( \frac{6x}{\pi} \right) \sin(nx) + \frac{2}{\pi} \int_{\pi/6}^{5\pi/6} \sin(nx) + \frac{2}{\pi} \int_{5\pi/6}^{\pi} \left( 6 - \frac{6x}{\pi} \right) \sin(nx) \]
\[ = \frac{12}{\pi^2} \int_{0}^{\pi/6} x \sin(nx) + \frac{2}{\pi} \int_{\pi/6}^{5\pi/6} \sin(nx) + \frac{12}{\pi} \int_{5\pi/6}^{\pi} \sin(nx) - \frac{12}{\pi^2} \int_{5\pi/6}^{\pi} x \sin(nx) \]

Knowing the following integrals:

\[ \int \sin(nx) = -\frac{\cos(nx)}{n} \]
\[ \int x \sin x = \frac{\sin(nx)}{n^2} - \frac{x \cos(nx)}{n} \]

Then performing the integration gives:

\[ b_n = \frac{12}{\pi^2} \left( \frac{\sin(nx)}{n^2} - \frac{x \cos(nx)}{n} \right) \bigg|_{0}^{\pi/6} + \frac{2}{\pi} \left( -\frac{\cos(nx)}{n} \right) \bigg|_{\pi/6}^{5\pi/6} + \frac{12}{\pi} \left( \frac{\sin(nx)}{n^2} - \frac{x \cos(nx)}{n} \right) \bigg|_{5\pi/6}^{\pi} \]
\[ = \frac{12}{\pi^2} \left( \frac{\sin(n\pi/6)}{n^2} - \frac{\pi \cos(n\pi/6)}{6n} - \frac{\sin(0)}{n^2} - \frac{0 \cos(0)}{n} \right) + \frac{2}{\pi} \left( -\frac{\cos(n\pi)}{n} + \frac{\cos(5n\pi/6)}{n} \right) + \frac{12}{\pi} \left( -\frac{\cos(n\pi)}{n} + \frac{\cos(5n\pi/6)}{n} \right) \]
\[ - \frac{12}{\pi^2} \left( \frac{\sin(n\pi)}{n^2} - \frac{\pi \cos(n\pi)}{n} - \frac{\sin(5n\pi/6)}{n^2} + \frac{5\pi \cos(5n\pi/6)}{6n} \right) \]
After expansion and elimination of the terms that cancel and the terms that equal zero, the above expression simplifies to:

\[ b_n = \frac{12}{\pi^2} \left( \frac{\sin(n\pi/6) + \sin(5n\pi/6)}{n^2} \right) \]

Substituting the expression for \( b_n \) and recalling that all the cosine coefficients equal zero, the Fourier series expression becomes:

\[ f(x) = \sum_{n=1}^{\infty} b_n \sin(nx) \]
\[ = \sum_{n=1}^{\infty} \frac{12}{\pi^2} \left( \frac{\sin(n\pi/6) + \sin(5n\pi/6)}{n^2} \right) \sin(nx) \]

Now consider the term \( \sin(n\pi/6) + \sin(5n\pi/6) \). By substituting increasing values of \( n \), the following table is obtained:

<table>
<thead>
<tr>
<th>( n )</th>
<th>( \sin(n\pi/6) + \sin(5n\pi/6) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>-1</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>-2</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>-1</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
</tr>
</tbody>
</table>

This table indicates that for any even values of \( n \), \( b_n \) will equal zero. For \( n = 1 \) or 5:

\[ b_n = \frac{12}{\pi^2 n^2} \]

For \( n = 3 \):

\[ b_n = \frac{24}{\pi^2 n^2} \]

For \( n = 7 \) or 11:

\[ b_n = -\frac{12}{\pi^2 n^2} \]
APPENDIX D. FOURIER EXPANSION OF TRAPEZOIDAL BACK EMF

And for $n = 9$:

$$b_n = -\frac{24}{\pi^2 n^2}$$

By substituting these values of $b_n$, the trapezoidal wave can therefore be described by the following Fourier series expansion:

$$f(x) = \frac{12}{\pi^2} \left( \sin(x) + \frac{2\sin(3x)}{9} + \frac{\sin(5x)}{25} - \frac{\sin(7x)}{49} - \frac{2\sin(9x)}{81} - \frac{\sin(11x)}{121} \cdots \right)$$

The third-order harmonic can be seen in the Fourier series expansion, with an amplitude of approximately 22% (two ninths) of the amplitude of the fundamental frequency. This is a large problem for delta-connected BLDC motors with trapezoidal back EMF, as third order harmonics will result in circulating currents within the delta.
STUDENT
Assessment Material

Copyright Permission Forms

[For use by Central Queensland University staff and students only]

Important Information for Staff

There may be occasions when you, as a staff member, require permission to use student material, originally authored by the student as part of the assessment requirement for a course of study at the University, for a specific purpose/s other than that for which it was initially intended.

The following Copyright Permission Forms have been designed for use on such occasions:

1. Request for Permission Form (prepared by the staff member making the request); and
2. Permission Authorisation Form (prepared by the student granting or denying permission).

Please ensure that the Permission Request Form is completed, and the student has completed the Permission Authorisation Form granting permission, prior to using Student Assessment material for ANY purpose other than that for which it was originally intended.

Important Information for Students

Under normal circumstances, as a student who is the sole author of assessment material produced while you are enrolled as a student at Central Queensland University ("University"), you will own copyright in that material.

If this is the case, and you are the Copyright Owner of the material, you will be in a position to grant, or deny, permission to use your material for the purpose/s described in the Permission Request Form you have received.

However, there are times when you may choose to rely on an exception available to you under the Australian Copyright Act (1968) e.g. fair dealing exception for the purpose of research or study (refer Student Brochure at: http://www.cqu.edu.au/students/brochure.pdf for more information), which will enable you to include other copyright material as part of your assessment, for which another person or entity is the copyright owner ("third party material").

On these occasions, you will NOT be in a position to grant permission for the use of your assessment material (for which you have previously chosen to rely on a fair dealing exception available to you as a student) for any other purpose than that which was originally intended, without obtaining written permission for use of any third party copyright material from the copyright owner/s of that material.

When granting any permission for use of your material, please also consider, as noted on the Permission Authorisation Form, that you may need to consider certain terms and conditions for the use of your material. For example, you may request acknowledgement in a particular way, or you may request a name/s, identifying features such as geographical details, be removed, or changed, to preserve confidentiality and anonymity.

*If you are unsure whether special circumstances may apply, seek advice prior to proceeding with any permission to use assessment material.

**Rather than relying on any exception, if you are aware of potential for use of your material beyond the initial assessment, you may choose to seek permission for use of third party material early.
1. Permission is sought by the following Central Queensland University Faculty/Division:

   Faculty of Sciences, Engineering and Health

2. To copy/communicate [as described in (4) below] the following Assessment material, authored by yourself as a student from the University, and for which you are the Copyright Owner.

   ♦ OR ♦

   To copy/communicate [as described in (4) below] the following Assessment material, authored by yourself as a student from the University, and for which you have written permission (attached) for use of any third party copyright material contained therein (which you would not otherwise be entitled to use).

   Technical Paper forming part assessment of courses ENEG14005 or ENTG13001.

3. Material is to be used for the following purpose:

   Electronic version in PDF format to be displayed on the Faculty website.

   (e.g. to use as an example for students studying (enter Course Name/Number)

4. For which the following details apply:

   | Time period (to a maximum of 2 years): One year |
   | Format/s: Electronic PDF format. |
   | Other relevant details: |

STUDENT RESPONSE FORM
### 5. Permission GRANTED:

I, as a Student from the University, and Copyright Owner of material described in the Request, **grant permission** for the University to copy/communicate the Assessment material for the stated purpose and period of time, as detailed in the Request (Section 4).

- [ ] No further action required; or
- [ ] Details of terms and conditions of use listed in Section (7) below.

Signed: ___________________________  Date: ___________________________

### 6. Student Permission, with Permission to use any third party copyright material that would otherwise not be possible, GRANTED:

I, as a Student from the University, **having obtained *permission (attached) for use of any third party copyright material, grant permission** for the University to copy/communicate the Assessment material for the stated purpose and period of time, as detailed in the Request (Section 4).

- [ ] No further action required; or
- [ ] Details of terms and conditions of use listed in Section (7) below.

Signed: ___________________________  Date: ___________________________

*Permission must cover use outlined in the Request, *e.g.* hardcopy use will not cover online use.

### 7. Terms and Conditions: [attach separate sheet if required]

**Attribution Statement:** [Are there any specific instructions for acknowledgement?]

[ ]

**Other:** ........................................................................................................

### Permission Denied:

I, as a student from the University, **deny permission** to copy/communicate the Assessment material as outlined in the Permission Request.

Signed: ___________________________  Date: ___________________________

Reasons for Denial (optional): ..........................................................................................
CONFIRMATION OF DEPOSITION

(To be appended to the thesis on submission of hard bound copy to the University Library)

PART A  
(Candidate to complete. Please print in black ink).

Candidate’s Name: Shannon Edwards

Degree: CF47 - Bachelor of Engineering (Co-op)/Diploma of Professional Practice (Engineering)

Thesis Title: Closed Loop Upgrade of a 5-DOF Robotic Manipulator

PART B  
(Office Use Only)

As at the date recorded below and in keeping with the Requirements for the Submission, Presentation and Deposition of Research Theses, the thesis described above has been deposited in the University Library.

_________________________________________  __________________________
Signature  Date

(Director, Division of Library, Information & Media Services (or nominee))

(The signed Confirmation of Deposition must be forwarded directly to the Research Services Office).
Author: Shannon Edwards

Dissertation Title: Closed-loop upgrade of a 5-dof robotic manipulator

Date: 1/6/2012

Author's Declaration:

Authors should note that unless specific prior written approval has been negotiated the following policy will apply:

Copies of up to ten percent (10%) may be made for the purposes of research or study with a view to publication.

If the above policy is not acceptable then the other options that are available subject to specific prior written approval are:

Copies of this dissertation in whole or in part may be made for the purposes of research or study with a view to publication and with the Librarian's permission.

Or

This dissertation may not be copied in whole or part without my permission.

Signed